BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

State Intellectual Property Office of the People's Republic of China

Title: Multi-step digital signature method and system			
Application Number:	96196055	Application Date:	1996.04.19
Publication Number:	1192834	Publication Date:	1998.09.09
Approval Pub. Date:		Granted Pub. Date:	
International Classification:	H04L 9/30 H04L 9/32		
Applicant(s) Name:	Satco Co., Ltd.		
Address:			
Inventor(s) Name:	F. W. Sudia		
Attorney & Agent:	cheng tianzhe		

Abstract

No abstract

[19]中华人民共和国专利局

[51]Int.Cl[6]
H04L 9/30
H04L 9/32

[12] 发明专利申请公开说明书

[21] 申请号 96196055.8

[43]公开日 1998 年 9 月 9 日

[11] 公开号 CN 1192834A

[22]申请日 96.4.19

[30]优先权

[32]95.6.5 [33]US[31]08/462,430

[86]国际申请 PCT/US96/05317 96.4.19

[87]国际公布 WO96/39765 英 96.12.12

[85]进入国家阶段日期 98.2.4

[71]申请人 塞特科有限公司

地址 美国纽约

[72]发明人 F·W·苏德亚 P·C·弗罗因德

S·T·F·瓦恩

[74]专利代理机构 中国专利代理(香港)有限公司

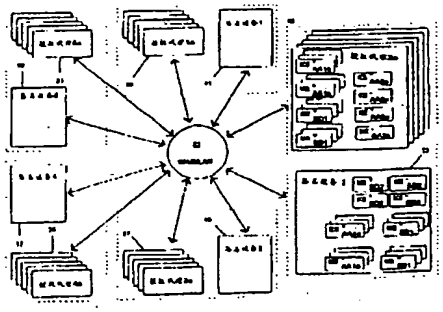
代理人 程天正 张志醒

权利要求书 1 页 说明书 34 页 附图页数 19 页

[54]发明名称 多步数字签名方法和系统

[57]摘要

多步签名系统和方法使用了多个签名设备(11, 13, 15, 17, 19)来加署单个签名，该单个签名可以使用单个公开确认密钥来确认。每个签名设备具有签名密钥的一子密钥，并响应从多个授权代理(23, 25, 27, 29, 31)来的授权而加署一个不完全签名。在一个串行的实施方案中，第一个不完全签名被加署之后，第二个签名设备对第一个不完全签名进行幂运算。在一个并行的实施方案中，每个签名设备加署一个不完全签名，且多个不完全签名被相乘以构成最后的签名。通过在多个签名设备加署签名的分布能力和在多个授权代理中加署不完全签名分布授权，使该系统的安全性得到加强。



权　利　要　求　书

1.一种数字签名方法，包括以下步骤：

产生私有签名密钥的子密钥；

在独立的电子签名设备中保存子密钥；

为签名设备确认多个授权代理；

对多个签名设备中的每一个，响应最少数量的授权代理的授权而将不完全签名加到电子消息中；

其特征在于，多个不完全签名组成了数字签名。

2.用于将数字签名加到电子文件上的系统，包括：

多个相互通信的签名设备，每个签名设备包括一个电子设备，该设备被编程以接收电子文件，并响应预定数的授权而用签名密钥子密钥附加不完全签名；

多个授权代理，每个授权代理可与相应的签名设备通信，每个代理包括一个编程的电子设备，以便向相应的签名设备提供授权。

3.用于在电子文件上加签数字签名的签名设备的连锁环系统，包括

签名设备的第一组，所述的第一组包括多个电子设备，每个电子设备被编程以便用来接收电子文件并为第一个签名密钥加上不完全签名，多个所述的不完全签名包括了第一数字签名；

签名设备的第二组，所述的第二组包括多个电子设备，每个电子设备被编程以便用来接收电子文件并为第二个签名密钥加上不完全签名，多个所述的不完全签名包括了第二数字签名；

其特征在于第一组包括了至少一个不在第二组中成员，且第一和第二组至少包括了一个公共成员。

4.代表使用一个电子密钥的电子方法，包括以下步骤：

将上述密钥保存在第一个电子设备中；

将电子代表证明传送给代表；

从代表处发送一个请求和代表证明给第一个电子设备；并

响应该请求和代表证明而用所述的第一个电子设备来使用电子密钥。

说　明　书

多步数字签名方法和系统

本申请是美国专利申请 Nos.08/181,859《具有密钥 ESCROW 特性的加密系统》以及美国专利申请 Nos.08/272，203《具有密钥 ESCROW 特性的增强加密系统和方法》的部分继续申请，这两项专利在此引入以作参考。

背景

公开密钥证明是由被委托的发布者签署的电子文档，被用来证明结合在公开密钥上的用户姓名和其它的相关数据。证明对公众提供了保证，由证明所识别的公开密钥，是由其名字列在证书中的用户所拥有的。描述公开密钥确认系统的主要的标准包括 ITU－T X.509 目录授权体制，以及美国银行家联合会 ANSI x.930-第 3 部分：对 DSA 的认证管理（草案）。许多实现方案包括了一个分级结构，其中每个被确认的发布者（被称为证明授权单位（CA））对下属的实体的密钥作确认。CA 以一种可验证（可以证明 CA 签署了该文件）但不可仿造（可以保证在高级别上确认即：除了 CA 外没有其它人签署了文件）的方式，将数字签名附加到电子文档。例如，在 CA 分级体系的最高级别，可以有相对较少的“根” CA，可能一个国家一个，由它们来认证子 CA。在分级体系的根 CA 以下，高级别的 CA（可能是银行）认证它们以下的较低级别的 CA（例如，公司），它们接下来签署单个用户的证明。

当 CA 产生了一个下面有许多用户的大的分级体系并使用它的签名密钥来签署高级别用户和子 CA 时，CA 的签名变得更加有价值。这样，CA 的签名密钥也就更可能成为从事经济利益的恐怖分子和犯罪分子以及从事经济刺探或通过信息战动摇经济的外国军事力量和间谍活动的目标。所有的这些情况也以同样的力度适用于用来签署货币的电子票据的密钥上。

迄今为止，对 CA 的私人签名密钥的安全性需要已经通过提供一个“认证签名单元”（CSU）提出，该 CSU 是一个防篡改的安全模块，满足联邦信息处理标准（FIPS） PUB 140－1，级别 3 或 4，该标准由美国商务部国家标准及技术局（NIST）发布。上述 CSU 在内部产生它的公开/私有签名密钥对，将私有签名密钥安全地且永久地“限制”

在设备的内部某个地区，无法从外部读取，而只输出相应的公开密钥，它可以被用来确认自己的签名．可以从 Boston,MA 的 Bolt,Batanek,和 Newman 得到的一个 CSU 被配置成为允许它的私有签名密钥备份版本用“K－of－N 门限”机制来产生，在该机制中，私有密钥被分成 N 份，并被放置在小的塑料数据－密钥中，每个数据密钥含有一个存储芯片．该数据－密钥是 Burnsville,MN， datakey 公司的专利产品．这样，如果 CSU 设备被破坏了，至少 K 个法定数量的数据－密钥可以重新构成私有密钥．

至少一个主要的安全标准组织，美国银行家联合会 ANSI X9.F1 关于批发银行应用的保密安全委员会推荐： CUS 应该是被设计成为能禁止任何私有密钥以任何形式从设备输出，从而防止任何可能的对密钥的未授权的偷窃和使用．这种方法将需要一个用于在灾难后恢复的精确的处理过程，包括同时对几个密钥对的使用．因为单个密钥只保存于单个位置的一个单 CSU 上， CSU 或存放位置的丢失将迫使 CA 使用另一个密钥对，才能继续业务．这就要求 CA 公布和/或秘密地分配一些（至少两个或三个）公开密钥，每个密钥由独立的码号（例如 BT01， BT02， BT03）来识别，以便用户可以继续认证签名，这些签名在一个 CSU（可能包括了 BT01 的私有密钥）被破坏之后由 CA 发布．参看 x.90-第 3 部分有关灾难恢复的处理过程．

发明概述

本方面的一个目标是提供一个数字签名系统（“签名系统”），它用于证书和其它高价值文件（包括契约，现金的电子票据，谈判文档，等），该系统具有提高的安全性和灵活性．

本发明的进一步的目标是提供一个签名系统，其中数字签名可验证地与签名密钥有关，且其中在文件签名操作中，单个签名设备不需要包含签名密钥．

本发明的进一步的目标是提供一个签名系统，该系统允许在维持有可用、未被泄密的签名服务的情况下，可以有一个或多个签名设备被丢失或泄密．

本发明的进一步的目标是提供一个签名系统，其中多个签名设备各自产生、修改、或合成一个或多个不完全签名，而多个签名设备的操作结果产生一个单数字签名．

本发明的进一步的目标是提供一个签名系统，其中多个授权代理直接或间接地授权每个独立的设备添加或修改不完全的签名。

本发明的进一步的目标是提供一个牢靠且易使用的机制，其中授权代理可短时地代表它们的授权能力。

这里阐述的多步签名系统使用了公开密钥加密方法来签署电子文件，这样，文件的接收者可用签名者的公开认证密钥来认证签名。与该公开认证密钥相应的私有签名密钥在普通签名操作期间的任何时间都不允许以完整和可用的形式存在在一个地方。相反，私有密钥包含“操作子密钥”，它可以被用来加署或修改一个不完全签名，多个子密钥的顺序操作产生一个签名，该签名可以用公开证明密钥来认证。没有所有或某些法定数量签名设备的签署，整个签名将不会完全。在进行签名处理之前，每个签名设备顺序地要求从所有或某些法定数量与它相关的授权代理处得到的授权。

如果在可操作子密钥的初始产生过程中，产生了一个完整的签名密钥，则在各子密钥被传播之后，完整的签名密钥被破坏。因为任何一个设备的丢失和泄密的危险性现在大大减少，每个签名设备的信息内容现在可以被复制（如，远程备份，或插入替代，或“热”备份），所以如果任何设备被破坏，它可以被替代（或重构），业务可很快继续。任何独立签名设备的子版本的影响都被降低级别，因为签名操作不可以用一个单独设备来完成。

一个多层的授权管理系统这样被建立，使得每个签名设备中注册了许多个体（或被指定的个体所使用的外部智能卡），而且参加签名操作的签名设备只根据法定数量的已注册的个体的授权而参加签名操作。这些法定数量的个体（被称为授权代理）也被要求对系统的变化进行授权，比如注册额外的授权代理、删除授权代理、改变对于签名设备可以执行的任何不同操作的法定数量要求、或者产生及分布额外的或替代的密钥集。

通过这种方法，一个签名便可以被使用了，它可以使用公开公证密钥来认证，但没有私有签名密钥存在于可能遭受泄密或异常灾难的单个位置。在中断签名业务之前或对手得到足够的信息仿造签名之前必需有多个位置发生差错或被泄密。各独立的签名设备不需要与使用单个完整密钥的 CSU 那样有高的安全要求。可以使用符合 FIPS 140-1 第 3 级标

准的相对较便宜的设备（即，一种防篡改设备），这样可以避免使用相对昂贵的第4级设备（当检测到篡改时，它进行主动措施来破坏或保护内部信息）的需要。

一种授权代表机制允许一个授权代理去让一个代表或法定数量的代表授权它的智能卡在短时期内加署上他/她的签名。

附图简述：

参考附图，本发明将在下文中被描述：

图1描述了依据本发明的一个可操作签名系统的基本结构的概要；

图2显示了一个具有签名设备的数据中心的优选结构；

图3描述了被授权代理使用的经确认的设备的优选结构；

图4描述了在系统启动和初始化的过程中用于临时确认未初始化的签名设备的过程；

图5描述了用于产生和分布全系统范围的授权密钥的可操作子密钥的过程；

图6描述了用于再确认签名设备的多步签名处理过程；

图7显示了用于确认和注册授权代理的整个系统结构；

图8描述了使用授权代理的多步签名处理过程；

图9描述了在例行多步签名操作过程中经过不同的授权代理和签名设备的文件流程；

图10描述了在例行多步签名操作中文件上签名的变化。

优选实施方案的详细描述

先讨论一些相关的数学处理过程，从而开始对多步签名方法进行最直接的解释。

A · 用序列的不完全签名相乘的方案

首先，属于“全系统范围授权”的公开/私有密钥对的一个保密的签名密钥“K_{SWA}"被表示为一定数量（“n0”）的子密钥（“a_i”），方法是签名密钥K_{SWA}可以作为任何门限定数（“t0”）的子密钥的乘积而被计算出，其中t0小于或等于n0。用这种表示方法后，当处理少于t0个子密钥时就比较困难或不可能恢复签名密钥K_{SWA}。这可以用以下方法来实现，例如：1）使用Shamir类型保密分摊机制（A Shamir，“如何分摊一个秘密”ACM通信，NOV.1979，v.22,N.11），2）使用Blakley类型秘密分摊机制（G.R.Blakley，“保护密码的密钥”国际

计算机会议录 1979，信息处理协会美国联合会， V.48,1979,pp.242-268）； 3）对密钥进行因子分解；或 4）产生一个密钥作为已知因子的乘积．所有的例子都要求私有密钥能够表示成为：

$$K_{SWA}=a_1*a_2*...*a_{t0}(\text{mod } 2N)$$

其中 K_{SWA} 是签名密钥，且 ai 是任意 t0 个子密钥的组合．

第二， 每个设备对前一个设备留下的不完全签名进行幂运算，而前一个设备使用私有密钥的一个子密钥 ai，这样通过使用多个设备，形成了签名．当使用“模 N”算法时，（其中的算法操作包括将结果用模数 N 相除，并将余数作为模 N 的结果），以下的指数乘积和连续幂运算之间的关系是成立的：

$$(x^{a1*a2})\ (\text{mod } N)=((x^{a1})^{a2})(\text{mod } N)=((x^{a2})^{a1})(\text{mod } N)$$

换一种叙述方式，如果底数值 x 用两个因子 a1 和 a2 的乘积作指数，结果相同于该底数先用第一个因子 a1 作幂运算，得到的结果再用第二个因子 a2 作幂运算．进一步地，幂运算的次序可以相反，所以如果该底数先用第二个因子 a2 作幂运算，得到的结果再用第一个因子 a1 作幂运算，结果也是相同的．这种关系可以发展到使用三个或多个因子的幂运算．除另有说明外，所有的算法操作都用模 N 运算．

在多步签名方法中，签名密钥的各个子密钥 a1,a2,...,ano 都被分布到各个设备．第一台设备通过将文件散列（符号“H”将用来表示散列操作的结果），将一个不完全签名加到文件上，并对该散列作幂运算：

$$\text{第一个不完全签名}=(H)^{a1}(\text{mod } N)$$

第二台设备用第二个子密钥 C 对第一个不完全签名进行幂运算，得到进一步的签名：

$$\text{第二个不完全签名}=((H)^{a1})^{a2}\ (\text{mod } N)$$

该过程不断重复，直到“t0”个设备用每个“to”独立子密钥对散列进行了幂运算，产生了最终的签名，它可以用公开密钥来 K_{SWA} 认证．

B．用异步的不完全签名相加的机制

另一种可选的实现类似的结果的方法，包括将签名权的私有密钥分为各个子密钥，这些子密钥可以被加起来（mod N）产生私有密钥．

$$K=a_1+a_2+...+a_t\ (\text{mod } N)$$

这就允许多步签名以异步的方式来执行，方法是，用每个子密钥对

散列作幂运算，独立地产生中间值（H）^{a1}，然后将这些中间结果相乘，如下：

$$S=H^{a1}*Ha^{2*}...*H^{a3} \quad (\text{mod } N)$$

这种方法比上述的顺序操作的方法有明显的操作上的优越性，因为它不需要将消息从一个位置引导到另一个位置。相反，一个中心的管理者可以以直接的方式，仅仅简单地将相同的消息（或散列）直接发送给每个用来作不完全签名的位置，然后将得到的不完全签名组合起来，产生最终的所需要的正式签名。这最后的合成操作不需要任何特殊的安全要求，因为它不加任何已包含在不完全签名中的信息，这就允许管理者从桌面设备上完成。实际上，不完全签名甚至可以方便地留待以后由作认证事件处理的接收者来组合！这给接收者以额外的处理工作负荷，但不会削弱正式签名的安全性。

建立在幂运算基础上的签名机制可以被修改，以允许多步签名，该机制包括：R.Rivest,A.Shamir 和 L.Adleman（“RSA”），“获得公开密钥密码系统的数字签名的方法”，（Communications of the ACM,v.21,n.2,pp,120-126,2 月，1978）；D.Kravitz,“数字签名算法”（“DSA”），美国专利 No.5,231,668;Desmet,Y.Frankel,“门限密码系统”，CRYPTO'89,PP.307-15,1989;Taher El-Gamal,“基于离散算法的公开密钥密码系统和签名机制”（“El-Gamal 签名算法”），IEEE Trans.信息处理理论，Vol,IT-31,NO.4,7 月 1985；S.Micali,“一种安全有效的数字签名系统，”MIT/LCS/TM-501 Massachusetts 理工学院，计算机科学试验室，4 月 1994；A.Menezes 等人,“椭圆曲线公开密钥密码系统”1993。

系统概况

图 1 阐明了与本发明一致的签名系统的结构概况。该结构包括多个签名设备 11，13，15，17，19，它们通过广域网或局域网互联起来。单个的签名设备 11，13，15，17，19 分散在 WAN/LAN 所允许的地理范围内，例如在分散的各大洲，分散的城市，或至少是一个城市的各个分散区域内。

在图 1 中，签名设备 2 将作为例子进行更详细的阐述。每个签名设备都被分配了一个永久的识别码（例如一个独一无二的序列码）和一个逻辑名字（例如“签名设备 X”），同时还有用于加密/解密通信的公

开/私有设备密钥对 12a， 12b 以及用于认证/签署签名的单独的公开/私有设备密钥对 14a， 14b．另外，每个签名设备都接收用于其余的签名设备的公开加密密钥 16 和公开校验密钥 18．

从这里往后，加密/解密密钥由“ KE ”表示，“ KS ”代表签名/认证密钥．一个加号“ + ”上标指示是公开密钥，一个减号“ - “上标指示是私有密钥．下标指示相应密钥对里的私有密钥的拥有者。

授权代理群 23， 25， 27， 29， 31 通过网络进行相互之间以及与签名设备 11， 13， 15， 17， 19 的互联．每一个授权代理是一个人，他通过一台委托的计算机设备（例如一个抗干扰智能卡，或是别的委托设备）进行操作，这一点将在下面进行更充分的讨论．授权代理可以分散在整个 WAN/LAN 21 的范围内，出于对签名系统管理的组织方便性考虑，可以假设授权代理群在大多数情况都靠近对应的签名设备．

图 1 中,授权代理 2a（项目 25 ）已经用实例进行了阐述，并采用与前面讨论过的、签名设备 2 所保留的密钥相同的密钥标记方式．每一个授权代理的委托设备被分配一个独一无二的名字，和用于加密/解密通信的公开/私有密钥对 20a,20b ，以及用于认证/签名的单独的公开/私有密钥对 22a,22b ．如果使用 RSA 公开密钥加密体制，那么签名和加密可以同时用一对密钥．授权代理也接收所有别的授权代理的公开加密密钥 24 和公开认证密钥 26 ．

签名设备也接收给所有授权设备的公开加密密钥 24 和公开认证密钥 26 ．类似地，授权代理的委托设备接收给所有签名设备的公开加密密钥 28 和公开认证密钥 30 ．

为了下面更容易解释多步签名过程，可以假设网络上所有的通信都采用公开密钥加密体制（ PKC ）进行加密，例如 RAS - 密钥 - 传输．还假设从一个网络实体送到另一实体的命令是由发送者使用标准（ PKC ）体制，例如具有 MD5 消息摘要的 RSA 签名体制签署的．在以后的图中，设备加密/解密密钥和设备签名/认证密钥都可被省略，但如前面所讨论的，这应该理解为它们存在于所有设备中．

图 2 显示了一个安全数据中心的计算机配置 48 的结构，图 1 中的每一个签名设备都可以在这张图上找到．除了签名设备 29 ，每一个数据中心配置 48 还包含一个独立的消息服务器 47 ．签名设备 39 专门用于签名操作并被放置于物理上安全的地方，例如密室．签名设备和外部

计算机网络之间没有直接连接。如下所列将提供给签名设备 39：一个用于多步签名 36 的共享密钥，签名设备自己的设备签名密钥，用于识别认证代理的表 18，一个公开认证密钥 40 的证明，一个匹配其子密钥 36 的公开密钥（这里的证明是通过多步方法由完全的 KS 分配的）。上述内容将在下面进行充分的讨论。

在多步签名过程中，签名设备 39 通过消息服务器 47 接收请求。消息服务器执行常规的通信过程，如去掉常规的、可能已经由中间过程署名的私有信封（服务器 47 不拥有签名设备的私有解密密钥），以及把那些在能被处理前就已提出的输入进行排队。消息服务器向签名设备提交签名消息，接收已签名（或部分签名）的结果，然后（a）将部分签名结果返回请求者，或者（b）将结果按协议中的路线发送到下一个设备。为了能接收和参与平常的通信协议，消息服务器也有一个公开/私有密钥对 32，33，用于它自己的消息签名，以及用于加密的公开/私有密钥对 34，35，以使它能接收和打开加密消息－从而使签名设备卸掉这个常规负担，而安全签名过程的安全性没有受到明显影响。

消息服务器 47 可以是在较低安全级别环境里的一个安全性能相对弱的计算机，例如在一个普通的安全数据中心里。消息服务器 47 与 LAN/WAN 相连，为签名设备 39 提供文档排队和通信服务。消息服务器 47 包括一个系统日志 49，用于维护一个往来于签名设备的消息和文档的跟踪审核记录。如所显示，签名设备和相关的消息服务器最好是分成两个物理上独立的计算机。虽然不推荐，但签名设备 39 和消息服务器 47 可以实现为高度安全环境里单个计算机上的独立的任务。

消息服务器也可以提供一层称为“防火墙”的保护，它在将所有事务性输入传递到签名设备之前独立地认证其有效性。否则一个“在线的”、可被公共网络访问的签名设备将面对无限制的破解尝试，以及旨在破坏服务的使网络饱和的攻击。破坏攻击会使日常证明发布崩溃，但对于依赖于先前的签名文档的用户（这是参加客户中的大部分），不会有削弱影响。然而，破解尝试会一直形成一个威胁，特别是当破解者发现了一些隐藏的缺陷时。消息服务器不仅能根据一个授权设备（签名设备和授权代理）的清单来认证所有的消息，而且可以采用更复杂的策略来识别攻击，在一定次数的失败尝试之后拒绝接入，并且执行复杂的步骤来跟踪错误输入数据源。这就允许签名设备的固件保持简单和易于生

效，同时也允许系统操作者修改其检查和躲避策略从而与现有的网络安全状态一致.

图 3 阐明了一个用于授权代理的工作站. 作为授权代理的操作员可以工作于安全性相对较低的地方，如办公室的桌面计算机或终端 51 . 每一个这样的计算机或终端 51 都有一个读卡器 53，每一个操作者都有一张安全的“智能卡” 55 . 每一张智能卡 55 都能安全地保存该卡的独一无二的私有解密密钥和私有签名密钥. 操作员可以使用智能卡发出签名指令. 这样一个委托设备可以使用 FIPS Level-3 设备完成，例如 Santa Clara,CA,National Semiconductor 公司的 iPOWER 卡，它可以在固件的级别上进行重新编程，以便在不替换物理设备的情况下完成新方法和安全的签名、授权过程的逐步演进. 每一个授权代理的委托设备至少有一个私有的签名密钥. 最好是生产厂商将私有签名密钥安装在设备中，同时，一致的公开认证密钥也由生产厂商“保证”. 这里的保证是指生产厂商已经在委托设备中包含了一个含有设备序列号、公开密钥、类型号以及另外的委托特征证据的电子消息，并且这个消息由生产厂商签署.

操作员使用他们的桌面计算机阅读和产生消息. 当操作员想给一条消息签名时，桌面计算机就将消息送到委托设备，委托设备使用设备私有签名密钥给消息附加一个数字签名. 在优选实施方案中，这个签名是一个二次签名密钥对的签名，这个二次签名密钥对是专门为特定的使用者而生成和鉴定的. 在这种方式中，系统可以继续使用设备的签名以便在任何指定事务中校验设备的委托级别，同时使用用户的签名来证明用户的身份和同意该事务. 这就使得用户密钥可以依靠各种各样的关于用户的身份和权限的管理信息来进行远端生成和废除，同时使得设备可以重用，或者掌管许多别的用户密钥对，用户也许希望将这些密钥对用于别的不相关的目的.

图 3 还阐明了一个授权代理使用的一个可能的委托设备的优选结构. 它包括一个封装在称作“智能卡”上的单个微芯片. 微芯片有用于电源和通信的输入/输出电路 42，和用于执行固件程序的微控制器 44 . 内存 52 包括用于操作微芯片的硬件的系统固件 43（类似于简单的操作系统）. 内存 52 还包括用于存储如下信息的区域：生产厂商安装的设备密钥 45；作为此处提及的协议的一部分接收到的用户密钥 47；用于

执行此处提及的协议的应用固件 49．另外没有用到的内存为临时存储提供工作空间 54．微芯片也可以包含一个可选的“加密单元” 46，这是一个特殊目的的算术加速器单元，它的硬件可以完成指数运算和其它一些加密/解密、签名过程中的算术运算．微芯片还包括一个可选的委托时钟 48（假定具有合适的电池电源），它由生产厂商初始化，可以用于时间标记签名．微芯片还包括一个可选的随机数发生器，用于加密/解密过程．智能卡也可以包含一个可选的噪声源（没有显示），例如二极管，在微芯片的内部或者外部，用于产生随机数．

先前显示在图 2 中的签名设备也可以是智能卡，它和授权代理的委托设备具有同样的通用设计．

网络中的设备应该按下列步骤进行初始化:

1、 加密密钥分配;

2、 签名设备临时证明;

3、 共享密钥分配;

4、 签名设备再证明;

5、 授权代理证明．

每一个将依次进行讨论．系统的初始化讨论之后，下面将解释签署高度保密的证明和其它文档的优选方法，同时还有进一步的变化和加强．

加密密钥分配

每一个签名设备和授权代理的智能卡都可以认为是“委托设备”，因为它是防修改的设备，它的功能只与所列的特征一致，生产厂商已经在其保护内存中存入了设备签名密钥对和设备加密密钥对．至少，这种设备的生产厂商可以证明在没有付出昂贵的修改努力的情况下设备不会泄漏它自己的或用户的私有密钥．每一个设备都有一个厂商签署的电子证明，它包括: 1）设备序列号; 2）设备的公开签名校验密钥; 3）设备的公开加密密钥．生产厂商可以安装两个独立的证明，一个用于签名确认密钥，一个用于加密密钥．签名设备使用公开/私有加密方案对其通信加密．另外一种替代方案是: 在没有厂商的证明时，通过给所有设备提供物理保护来实现这种方法，例如，在安全的密室，用小型（笔记本）电脑代替委托签名设备来引导初始化任务．

假设每一个委托设备都从一定的基本功能开始，例如由软件检查是

否有能力进行初始化并且能够通过网络或电子邮件系统接收消息，以使设备与其它委托设备进行通信．同时假设至少存在一个签名设备指定为“领导”设备，它能够从负责初始化系统的人工操作员那里接收关于系统初始化的信息．

准备系统的下一步是为设备交换设备密钥．下面是密钥分配过程:

1）指定为“领导”的签名设备从人工操作员那里接收系统别的签名设备的识别码．领导设备把它的公开加密密钥和公开签名校验密钥发送给其它的签名设备．领导设备也可以送一条消息以认证正由它操作的固件的有效性，例如通过把固件进行散列，用自己的设备签名密钥给散列值签名并将它送达别的设备．

2）当另一个签名设备收到领导设备的公开加密密钥后，每一个别的签名设备都将自己相应的公开签名校验密钥和公开加密密钥证明送回给领导设备．如果领导设备送的是它的固件的散列信号，则其余的签名设备都将自己的固件散列，然后比较两者的结果．两个散列值必须匹配，否则各个签名设备停止加入协议并通知操作人员．杂散值的比较保证所有签名设备使用相同的固件，这可以检查一个领导设备是否是一个“impostor”．每一个签名设备可以任选地返回各自固件的散列值给领导设备．

3）领导设备将各个固件的散列值与自己的相比较，来检查各个设备中是否存在“impostor”．

现在所有的签名设备都已经接收到了别的设备的公开加密和签名校验密钥．可以这样理解: 今后所有的消息都将用发送方的私有签名密钥进行签名，接收方用发送方的公开校验密钥进行校验．还应该这样理解: 所有的通信都将用接收方的公开加密密钥进行加密，然后用接收方的所有解密密钥进行解密．

这些附加的签名密钥在多步签名中没有使用（将在下面讨论），但用于网络实体间常规通信的加密和签名，以此作为一个设备独立身份的证明．当生成和分配在实际多步协议中使用的主密钥的子密钥时，这种在群里的身份和资格的证明就非常重要．

签名设备临时证明

图4阐明了未初始化的签名设备的临时证明．在这个过程中，签名设备的公开密钥证明（由生产厂商签署或未被签署）会被临时管理者

（“管理者”） 61 签署的证明所替代. 管理者最好是个负责系统初始化的操作员，通过管理者的个人智能卡进行操作. 当签名设备（属于目标群）为多步签名生成签名密钥时，这种临时证明在它们中间建立起一个提升了的安全等级. 在实际使用中，希望临时管理者在操作时有多个证人在场以保证正确的步骤，临时证明只在必要的短时间内有效（几分钟或最多几小时）以执行完整的主密钥的生成协议.

临时证明进行如下:

1）管理者 61 生成一个私有签名密钥 63 和一个相应的公开校验密钥 65.

2）管理者 61 把公开签名校验密钥 65 传给每一个签名设备 11，13，15，17，19.

3）每一个签名设备 11，13，15，17，19 生成一个私有签名密钥 67，69，71，73，75 和一个公开校验密钥（没有显示），把签名密钥证明请求送给管理者 61. 这个签名密钥证明请求是一条电子消息，它包含签名设备的名字（例如，一个设备的序列号和/或逻辑名字、如“SD1”）、设备新近生成的公开签名校验密钥、和别的期望的管理信息.

4）管理者用管理者的私有签名密钥给每一个证明请求签名.

5）管理者返回已签署的签名密钥证明 68，70，72，74，76 给各个签名设备 11，13，15，17，19. 签署的证明 68，70，72，74，76 用带有正确下标的公开签名密钥（KS +）的符号进行描述，下面附加上管理者的签名（“--ADMIN”）。当然，这种证明还包括设备标识和类型的信息.（没有显示）

6）签名设备之间互相交换新的临时公开签名校验密钥证明.

每一个签名设备现在都拥有: a）管理者的公开校验密钥; b）自己的临时私有签名密钥; c）自己的临时证明，它由管理者签署并载有签名设备的临时公开签名校验密钥; d）别的签名设备的临时签名校验密钥证明. 每一个签名设备都可以用管理者的校验密钥去校验加在从别的签名设备收到的临时证明上的管理者的签名.

每一个签名设备现在可以使用由临时管理者证明的签名密钥来交换消息，而向更稳固控制的协议阶段前进. 为便于解释，可以假设在涉及多签名操作中的网上通信时，从这个位置到设备重认证的结束，都用

短期管理者确认的签名密钥签署，而每个接收者确认发送者的发送者签名．如果一条消息没有被正确地签名，那么这条消息将被拒绝，协议将不能继续除非提供一条相符的消息．更进一步的打算是：当在多步初始化和签名操作中收到一条不正确的签名或未签名的消息，就应该执行某种形式的征兆分析或征兆反应．

授权代理临时证明

图 4 阐明了授权代理的临时证明．一个签名设备仅仅在法定多数应答授权代理的授权要求时才附上一个部分签名，这将在下面进行全面讨论．在临时管理者授权下运行的签名设备也要求法定数量的授权代理．授权代理的临时证明保证只有指定的代理人才能在初始化过程中授权签名设备．

授权代理的临时证明与上述的签名设备的临时证明很相似，具体过程如下：

1 ）管理者 61 把公开签名校验密钥 65 传给每一个签名设备 23，25，27，29，31．

2 ）每一个授权代理生成一个私有签名密钥证明请求，它至少包含以下信息：a ）授权代理的名字（人的可区分的名字）；b ）代理的委托设备的标识码（如智能卡的序列码和类型码）；c ）代理人的签名校验密钥；d ）代理人的委托设备的签名校验密钥（可以作为委托设备是已知类型的保证）．

3 ）管理者用管理者的私有签名密钥给每一个证明请求签名．

4 ）管理者返回已签署的签名密钥证明给各个授权代理设备

子密钥分配

图 5 阐明了系统范围授权（ SWA ）“正式的”签名密钥的“可操作子密钥”的生成和分配．一个签名设备（这里是签名设备 5（物件 11 ））设计为领导设备．操作员至少要给领导设备提供如下信息：

a) 将一个密钥分成子密钥的门限参数，如生成的总数和要加署 SWA 签名的最少个数．

b) 需要分配给公开/私有密钥对的识别号码和/或逻辑名称，例如密钥序列号“ KS - 01234 ”，或逻辑名称“ SBT01 ”．

c) 需要赋给各自子密钥的识别号码和/或逻辑名称，例如“ SWA - SHR - 56789 ”，或“ BT01a ”

d) 授权代理的设备证明，这些授权代理最初被允许为每个设备授权特殊的签名。

操作员还提供一个用来限制签名设备中片段（ fragment ）总数的数量，它在当单个签名设备有多个主密钥时被使用，这将在后面详细讨论。

下一步是生成签名密钥的子密钥，叫做“系统范围授权”密钥（ SWA ），它用于管理系统。公开的 SWA 签名密钥和相应的私有 SWA 子密钥按如下步骤生成和分配：

1) 每一个签名设备 11， 13， 15， 17， 19 给领导签名设备 11 发送一个加密的随机“种子”信息。

2) 领导设备 11 组合种子信息，并用它生成一个公开的系统范围授权签名校验密钥 91（ $KS_{SWA}+$ ）91，最终用它来校验正式签名。

3) 领导设备生成私有 SWA 签名密钥的可操作子密钥 92， 95， 97， 99， 101 。首先使用已知的密钥生成方法来生成私有/公开密钥对，然后可以使用许多种已知的私有签名密钥分裂方法中的一种，来把私有签名密钥 92 分成子密钥。子密钥的生成要记住一个要求：要有足够的最小数目 n0 个独立子密钥来完成系统范围授权签名。

4) 领导设备 11 把 SWA 公开校验密钥 91 和一个私有签名子密钥 95， 97， 99， 101 发送给每一个签名设备，同时给自己保留一份 SWA 公开校验密钥 91 的拷贝和 SWA 私有签名密钥 93 的一个子密钥的拷贝。每一个 SWA 私有签名子密钥随如下附加信息一起传送：

a) 标明该密钥是一个签名子密钥的类型码（同时指示子密钥长度）；

b) 用于 SWA 公开校验密钥的独一无二的识别码；

c) 用于每一个相应的 SWA 私有签名子密钥的唯一的识别码；

d) 已分配的 SWA 私有签名子密钥总数；

e) 用于完成一个 SWA 签名的 SWA 私有签名子密钥的最小个数；

f) 接收别的 SWA 私有签名子密钥的签名设备的标识；以及

g) 授权代理的证明，该授权代理最初被允许在目标签名设备识别上授权使用 SWA 私有签名子密钥。

领导设备 11 用各个签名设备打算使用的、经过鉴定的公开加密密钥给每一个 SWA 私有签名子密钥加密。

5）领导设备 11 给操作员输出公开 SWA 校验密钥并删除如下信息:

a、 整个私有 SWA 签名密钥（如果在生成过程中存储了整个密钥的话）; 以及

b、 全部 SWA 私有密钥的子密钥（留给自己使用的那个子密钥除外）.

6）每一个签名设备接收方将 SWA 私有签名子密钥和初始授权人的证明存在一个防修改的内存区域.

最好是私有 SWA 签名密钥至多只在领导签名设备中存在并且只在用来生成和分配子密钥的最短时间内存在. 这样, 整个的 SWA 私有密钥的存在就不是为操作之用的, 而只有在生成过程中的一小段时间内是易受攻击的.

在这个阶段, 每一个签名设备都安全地获得了: a)一份公开 SWA 签名校验密钥的拷贝; b)一个私有 SWA 签名子密钥.

为了在下面的讨论中解释一个例子, 可以假设（为了简化）需要签署 SWA 签名的子密钥的最小数目 n0 为五个密钥中取二个. 应该理解, 也可以选一个较大的数, 很可能至少是三个, 这会增加系统的安全性能, 但也会增加签名过程的步骤.

签名设备的再证明

在先前初始化协议的步骤中, 一个临时管理者 61 在临时管理者 61 的授权下为设备签名校验密钥做了证明, 并用管理者的临时签名密钥给签名设备证明签了名. 在再证明的过程中, 每一个签名设备将它自己的公开密钥的新的再证明请求送到别的签名设备中循环, 这些设备都要使用多步签名以在 SWA 密钥下获得证明.

图 6 阐明了对签名设备 1 再证明的步骤. 别的签名设备通过重复这一过程来完成自身的再证明. 对于签名设备 1 的过程如下:

1) 签名设备 1 生成一个无签名的证明 103 并将它发往签名设备 2. 这个证明至少包含以下信息: a、签名设备的身份标识（例如, 序列号和/或设备逻辑名）; b、用于设备签名密钥的公开签名校验密钥. 需要再证明的密钥就是在协议开始时由设备生成的初始公开密钥, 并且开始由管理者临时证明. 这个密钥现在要成为设备的永久标记, 用来标记它在掌握特定 SWA 密钥的子密钥的签名设备群里的成员地位. （在这个

过程中，设备签名密钥和相关的生产厂商的证明都保持不变，并将作为设备的原始和基本特征的证明而永久地保留下来。）

2）签名设备 2 用它的 SWA 签名密钥 93 附加一个部分 SWA 签名。部分 SWA 签名通过两步形成。首先，签名设备 2 应用一个“散列”函数（如 MD5 或 SHA）来生成一个减长字符串，可以证明该字符串对于非散列的证明是相关的。这个字符串表达为二进制形式，这样就可以作为一个数值（大的整数）来处理。第二步，签名设备 2 用散列字符串与它的 SWA 签名子密钥的乘幂来构造一个部分签名。即签名设备 2 用如下公式，计算出一个数值，把它变成部分签名：

$$-- SD2 = (HASH(CERT))^{[KEY\ SHARE\ 2]} \text{ modulo } N$$

（注意在文字和图片中，往往通过在签名者的标识符前面加一个长的破折号来指示一个构成签名块的比特串。所得结果块往往加在已签明的数据块的底部，要不然就可以由上下文明显看出。）

3）签名设备 2 将部分签署的证明 105 送往签名设备 3。

4）签名设备 3 通过已经应用过的部分签名--SD2 的乘幂来完成系统授权签名。签名设备 3 用下列公式计算一个数值：

$$-- SD3 = [-- SD2]^{[KEY\ SHARE\ 3]} \text{ modulo } N$$

$$=((HASH(CERT)\text{exp KEY SHARE 2})\text{exp KEY SHARE 3})$$

$$=-- SWA$$

签名设备 2 签署的部分签名可以作为审核后缀而允许保留在文档里。注意在这个简化的例子里只要求 2 个部分签名。

5）签名设备 3 向签名设备 1 返回已签署的证明，签名设备 1 将这个证明的拷贝发送到别的签名设备，因此使它们能够校验它以后的签名。

这个例子中，签名设备 2 和 3 按这一顺序附加签名。任意组合的签名设备可按任意顺序进行签名，从而可以得到相同的签名（只要数目超过最小值 t0）。

再证明是重要的，因为将来整个系统签名设备执行的操作只会被作为对那些被 SWA 签名证明过的设备（如下面描述的授权者的设备）所提出的请求响应。签名设备本身也可以对别的签名设备提出请求。由这个过程，这些签名设备通过使用此处定义的多步签名处理使自己作为整体而成为首先被系统范围授权所证明的设备。

在一个先前的再证明处理的替换方案中，签名设备群可能在领导设

备生成初始密钥前提出它们的再证明请求（无签名证明）．领导设备将在分割其为片段和删除整个密钥之前生成 SWA 私有签名密钥的时候签署这些证明，这样做看来没有任何优势，因为系统的主要功能就是用高度受控然而有效的方式给证明签名．

授权代理的再证明

图 7 和 8 阐明了授权代理的认证和登记步骤．图 7 显示了一个总的系统结构，而图 8 则阐明了一个证明请求的处理顺序．签名设备将给授权代理的证明附加一个正式的系统范围授权签名，从而为每一个授权代理认证一个公开签名校验密钥．在登记过程，每一个签名设备都将更新一张内部存储的特定授权代理的表，这些授权代理有权指示签名设备应用其部分签名．在日常操作中，只有当请求被最小数目的、被临时证明的或被 SWA 证明的授权代理签署时（或是收到最小数量的单独被签署的消息时），签名设备才会附加其部分签名，这将在下面详细讨论．下面列出一个认证授权代理 3a(AA3a)和在签名设备 3 上登记 AA3a 的例子．

出于讲解的目的，可以假设签名设备 3 和 1（图 7，项目 15 和 11）是 5 个签名设备中选出来附加 SWA 签名中的 2 个．

1）授权代理 3a 通过 LAN/WAN 21 向签名设备 3 提交一个为自己作再证明的请求．（或者，授权和/或登记可被限制为通过有限的访问通信通道而向签名设备作直接输入，例如，和一台单独的个人电脑的直接连接）．认证请求至少包括下列信息：a、授权代理名字（可以区别的人名）；b、代理的委托设备的身份码（例如，智能卡序列号和型号）；c、给个人代理使用的签名校验密钥（初始时由临时管理者签署）；d、给代理的委托设备使用的签名校验密钥，这用来保证他的设备是已知类型．当所有的或大体上所有的操作在广泛分布的地域里进行，使系统操作员不可能通过直观的检查去认证一切时，这一点特别重要．

2）签名设备 3 给证明 121 附加一个部分 SWA 签名（－SD3），并将这个经过部分签名的证明 123 送给另一个签名设备．

3）签名设备 1 授权同意部分证明可以送往 SDI．

4）签名设备 1 用它的 SWA 签名子密钥 93 完成签名处理．

5）签名设备 1 向签名设备 3 返回完全签名的证明 125．

6）签名设备 3 保留一份已签名的证明 111 的拷贝，把 AA3a 登入

授权代理的日志 113，将已签署的证明 125 返回给授权代理 3a．

所有准备登记到签名设备 3 上的授权代理 101 都会重复这个过程，给每个授权代理 101 留下一个已签署的证明，给签名设备 3 留下一个包含所有证明的日志 113．对于别的签名设备 11，13，17，19 的所有授权代理都重复这个过程。

多步签名

在这个阶段，签名设备已经用 SWA 私有签名密钥的子密钥进行了初始化．签名设备已经对自身进行了再证明，授权代理已经进行了再证明并已登记了各自的签名设备．现在系统已经准备好进入到系统管理和提供正式的认证功能的日常服务中．在接下来的讨论中，将说明用系统范围授权密钥的多步签名，这是在系统管理中很典型的应用．正如将要讨论到的，和系统授权密钥一样，在同一族设备的多步签名中会生成和使用附加的“主”密钥，区别在于用这种主密钥签名的消息内容从本质上看是不可管理的．

图 9 和 10 阐明了使用系统范围授权密钥的多步签名．图 9 阐明了从各种授权代理和签名设备出来的文件流，图 10 阐明了在文件上的签名进展．这个例子假设授权代理 1a 和 1b 授权签名设备 1 附加一个部分签名，授权代理 2a 和 2b 授权签名设备 2 完成一个 SWA 签名．为简化，我们假设为激活每个签名设备需要任意两个授权代理．处理顺序如下所示：

1） 授权代理 1a 通过 LAN/WAN 收到一个签名请求．请求是一条具有消息头 133 和要签名文件 135 的电子消息 131．消息头含有一个命令码，指明该消息是一条签名请求。

2） 授权代理 1a（图 9，项目 132）去掉消息头并执行一系列程序上的检查来决定文件是否应该签名．特定的程序上的检查（它可以包括人工操作员 AA1a 的判断，且会根据文件执行的目的而变化）并不是与多步签名处理本身密切相关的．当确认文件可以被签名时，授权代理 1a 就用代理的秘密签名密钥给文件签名（密钥是由 SWA 签名再证明过的）．如图 10 显示，授权代理 1a 的签名（--AA1a）由这个文件散列后的散列序列与 AA1a 的秘密签名密钥的乘幂决定．然后 AA1a 附加一个新的消息头并将已签名的证明 137 送给授权代理 1b（这是与授权代理 1a 一样的、用于同一签名设备的另一个代理）．

3） 授权代理 1b（图 9，项目 138）去掉消息头并执行一系列程序上的检查（并不是与多步签名密切相关的）来决定文件是否该签名。当确认证明应该被签名时，授权代理 1b 也给文件签名。如图 10 所示，AA1b 的签名（--AA1b）由下列决定： a、把文件和 AA1b 的签名的级联结合体散列； b、用 AA1b 的签名密钥与散列序列作乘幂运算。AA1a 的签名留在文件中作为审核信息。AA1b 附加一个新的消息头并将两次签名的文件 139 送往签名设备 1（图 9，项目 11）。

4） 签名设备 1 收到两次签名的文件 139，去掉消息头，并校验文件是否承载了来自它上面已登记的授权代理签署的足够数量的签名（本例中是 2）。如果是，签名设备 1 去掉授权代理的签名并附加一个部分 SWA 签名。如图 10 所示，部分 SWA 签名（－ SD1）由散列的基本文件序列（不包括授权代理签名）与签名设备 1 的 SWA 签名子密钥 93 的乘幂决定。然后签名设备 1 附加一个新的消息头，把部分签名的文件送往另一个签名设备的授权代理，这里是签名设备 2 的授权代理 2a。

5） 授权代理 2a（图 9，项目 143）去掉消息头并执行一系列程序上的检查（并不是与多步签名密切相关的）来决定文件是否应该签名。当确认证明应该被签名时，授权代理 2a 给文件签名。如图 10 所示，AA2a 的签名（--AA2a）由下列决定： a、散列证明和部分 SWA 签名（－ SD1）的级联结合体； b、用 AA2a 经过再证明的签名密钥与散列序列作乘幂运算。SD1 的部分 SWA 签名留在了文件里。然后 AA2a 附加一个新的消息头并将签名的证明 145 送往授权代理 2b（图 9，项目 147）。

6） 授权代理 2b（图 9，项目 147）去掉消息头并执行一系列程序上的检查（并不是与多步签名密切相关的）来决定文件是否该签名。当确认证明应该被签名时，授权代理 2b 给文件签名。如图 10 所示，AA2b 的签名（--AA2b）由下列决定： a、散列证明、部分 SWA 签名和 AA1a 签名的级联结合体； b、用 AA2b 经过再证明的签名密钥对散列序列作乘幂运算。部分 SWA 签名和 AA1a 的签名留在了文件里。然后 AA2a 附加一个新的消息头并将签名的证明 149 送往签名设备 2（图 9，项目 13）。

7） 签名设备 2 收到已签名的文件 149，去掉消息头并校验文件是否承载了来自它上面登记的授权代理签署的足够数量的签名（本例中是

2）．如果是，签名设备 2 去掉授权代理的签名并修改部分 SWA 签名以完成 SWA 签名．如图 10 所示，完整的 SWA 签名（－ SWA）由签名设备 1 附加的部分签名（－－ SD1）与签名设备 2 的 SWA 签名子密钥 95 的乘幂决定．然后签名设备 2 附加一个新的消息头，把部分签名的证明 151 送往 AA1a（最先的授权代理）．

在上面描述的例子中，要用两个签名设备来附加系统授权签名，每个签名设备要求获得来自两个授权代理的授权．可以在生成子密钥时，调整系统中用于完成一个签名所需的签名设备的总数，每个签名设备所需的授权代理的门限数值也是可调的．例如，可以要求 5 个签名设备中的 3 个来完成系统授权签名，每个签名设备所要求的授权代理的数目可以各不一样，这依赖于人们对所需系统安全目的而考虑的级别．

按上面所讨论的过程建立一个多步签名以后，要采取一定的核心管理措施，这要以系统范围授权存在的情况下所授权的别的签名设备法定多数“同意”为条件．下面讨论一些管理措施．

为实行这样的管理和决定，每一个抗篡改签名设备内的固件要编程成只对按如下签名的命令作出响应：

1、 对于部分签名请求，由合适的法定数的授权代理签名；

2、 对于系统管理的改变，系统范围授权自己签名．

这就是说，在一个更好的实施例里，除非法定多数的全部签名设备的法定多数授权者同意，否则在一个授权者的清单或任何一个签名设备的相关要求都不可能作出任何改变．有些例子里，为了一个很小的修改，例如授权执行加密的备份，也要取得全系统的同意被认为是过度繁琐．但是，这种管理性的改变可以期望是相对少和不频繁，与大量的正式事务和系统安全要求相比，在任何情况下都应该获得这种同意．注意在这个例子里，只要求 4 个人的签名来（在）证明和（在）登记一个用户．

并行签名

图 11 阐明了在多步签名系统中并行实施例的文件流．在这个图解中，可以假设系统总共有三个签名设备 169a， 169b， 169c，并且三个签名设备都被要求完成 SWA 签名．应该理解，并行签名是可以做成适合于不同数量签名设备的．

在并行方法里，一个文件协调者 161（“协调者”）接收到一个要

签名的文本 163 。协调者可以是但并非必要是一个签名设备的授权代理，在这里为了广泛性而把它图示为一个独立的实体。

文件协调者 161 产生三个要签名的文本 163 的拷贝 165a， 165b， 165c （或是文件散列以后的拷贝）。每一个拷贝送到第一个授权代理 167a， 167b， 167c，接着送到第二个授权代理 171a， 171b， 171c，然后送到第三个授权代理 169a， 169b， 169c，最后返回协调者 161。由下面更全面的讨论知道，文件协调者把三个签名设备的独立签名结合起来，产生一个系统范围授权密钥（ - SWA ），并把这个密钥附加在原来的文件 163 里以产生一个签名的文件 173。

图 12 阐明了其中一个拷贝的处理，以及将三个部分签名结合为一个系统授权签名。应该理解，每一个拷贝经历相同的操作，不同的是各个授权代理和签名设备会根据各自不同的签名密钥附加签名或部分签名。

在这个例子中，要求有两个授权代理授权它们相应的签名设备 169a 附加签名。协调者 161 送要签名的文件的第一个拷贝 165a 到第一个授权代理 167 去签名，同时还送出路由和信息头（未示出），授权代理 167 附加它的签名（--AA1a ）并将已签名的拷贝送到第二个授权代理 171a。第二个授权代理 171a 附加上第二个授权签名并将（两次签名过的）文件 179a 送到签名设备。签名设备 169a 校验两个授权签名，在这个拷贝上附加它的部分签名（ - - SD1 ），将签名拷贝 181a 送回协调者 161。

另外两个签名设备（没有显示）附加部分签名给要签名的文件的拷贝并将已签名拷贝 181b， 181c 送回协调者。三个拷贝可以并行地处理。

在协调者收到已被签名的文件的三个拷贝后，协调者将三个部分签名（--SD1,--SD2,--SD3 ）相乘。三个部分签名的结果就是系统范围授权签名（ - - SWA ）。

签名设备和授权代理的智能卡应该是受托设备。这种并行的多步签名方法的安全性能不依赖于协调者工作站的物理安全性能。协调者不必处理任何用于授权签名设备的秘密密钥（虽然它有路由加密和用于私有性和身份识别目的的签名密钥）。

协调者的功能可以在授权代理之间扩散。第一个授权代理可以接收欲签名的原始文件并指定另一个授权代理（或者甚至可是另一个非授权

代理的实体，例如签名设备中的一个服务器）来接收和合并部分签名。可以期望这种结构的正常操作如能使协调者不仅接收要签名的文件，还负责转发签名的文件到最终的接收者，则使它更为可取。

添加/删除授权代理

每一个签名设备都有一个相关联的授权代理群。因为人们可以进入或退出有组织，系统应该包括这样的措施以便通过添加或删除授权代理的委托设备的公开密钥来给系统提供动态地添加或删除授权者。添加或删除一个授权代理是通过提交一个添加或删除代理的公开密钥的命令给签名设备来实现的。这个命令采用电子消息的方式，包括一个添加/删除命令码，附加信息（在下面讨论）和授权签名。

授权签名可以来自同一签名设备上别的授权代理，添加/删除处理可由单个签名设备在本地完成。在可替换的一种方案里，添加/删除过程要求获得系统范围授权密钥的签名，因此要求获得在相关的签名设备上的法定多数授权代理对这个改变的同意和授权。在另一种可替换方案里，不同的授权代理有不同的能力，能力强的授权者在系统范围授权密钥下进行添加或删除，能力弱的授权者可以在本地群的授权下进行添加或删除。更好的是，授权代理的添加和删除都要求系统范围授权密钥的签名。

图 13 阐明了用于删除一个授权代理的命令 201 .命令 203 （ 201 or 203)的附加信息包括： a、代理的名字 205； b、代理的称号 207； c、授权代理的 ID 号 209，代理将从它那里被删除； d、与要删除的授权代理相联系的委托设备的身份码 211 。当收到一个正确的签名命令后，签名设备从它的内部授权代理的清单上将授权代理的公开校验密钥删除。

图 14 阐明了用于添加一个授权代理的命令 213 。命令的附加信息包括： a、代理的名字 217； b、代理的称号 219； c、授权代理的 ID 号 221； d、一个表明授权给代理的权利的管理等级 225； e、新的代理的授权的超时日期 223； f、身份码 227，用于主密钥或授权代理指示签名设备应用的密钥； g、代理的委托设备的 ID 码 229； h、带有委托设备的公开签名校验密钥的证明 231 。最好是新代理的公开密钥在 SWA 签名密钥的授权下已经证明过了，且在这个命令中包含了这个证明。与授权代理相关的委托设备的由生产厂商签署的设备证明 231 也包

括一个保证，即授权代理的私有签名密钥永久地限制在智能卡中或别的已经证明具有最小程度的安全特性的委托设备中．（最好是设备的最小程度的安全特性也包含这样一个事实：智能卡与人类使用者的联系采用生物信息。例如，生产厂商可以指出除非使用者动作一个连接的指纹阅读器，匹配的指纹数据存储在卡内并用来激活它，否则卡不能创立它的用户签名）．收到一个正确签名的信息后（例如，在完成 SWA 多步签名后），签名设备就把新的代理的信息加到它内部的授权代理的清单中．

添加/删除卡生产厂商和型号

正如上面所讨论的，授权代理通过委托设备起作用，委托设备可以是生产时就决定了其安全特性的智能卡．在添加一个授权代理的情况下，代理的委托设备必须是一种经过核准的型号．在系统初始化时，可以被系统接收使用的委托设备的型号被输入．一段时间以后，新的型号变成可用，安全过程得到加强，使得老的型号不再被接受．所有签名设备都在内部维护一张可以接受的型号表．

当在签名设备间循环传输一个新厂商的电子请求时，就可以加入新的厂商信息。图 15 阐明了一个请求的例子．请求包括命令 243，生产厂商的名字 245，型号或编码 247 和一个公开签名校验密钥 249，共同结合在由系统范围授权密钥签署的消息 241 内．

老的生产厂商可以通过循环传输由 SWA 密钥签署的电子请求使它被删除掉，即从签名设备的表里移走生产厂商的公开校验密钥．图 16 阐明一个例子，请求 251 包括命令 253 和生产厂商名字 255．这些添加/删除的请求，一旦被法定数量的设备签署，就会被发送到所有设备，那些设备用 K^{+}_{SWA} 进行校验并产生操作．

已经被核实的生产厂商的新的型号可以通过提交一个由 SWA 密钥签名的加入新型号的电子请求从而加入一个新型号．图 17 阐明了一个请求 261 的例子。它包括命令 263 和生产厂商名字 265，型号 267 和一个由生产厂商签署的证明 269，说明该特殊的型号适应一定的安全标准．（例如，证明某个型号满足 FIPS 第三级的要求）．

老的型号可以通过提交一个由 SWA 密钥签名的电子请求，以便从签名设备的表中移走型号的方式使老型号被删除．图 18 阐明一个请求的例子 271，包括一个命令 273，生产厂商名字 275 和型号 277．

添加/删除签名设备

一段时间以后，我们会希望从系统中添加或删除签名设备．每一个签名设备都保留了一张系统内别的签名设备的表，这些签名设备含有SWA 密钥的子密钥（或者是将在下面讨论的、用于多步签名的别的主密钥的子密钥）．每一个签名设备的标识由如下定义：1、设备识别号码（如序列号）；2、识别公开校验密钥（由生产厂商安装，并由生产厂商签名证明；或者是由SWA 签名进行再证明的类似密钥）；3、设备公开加密密钥（用于给设备发送加密消息）；4、任何后来证明的、唯一拥有的公开密钥．

新的签名设备通过在别的设备间循环传输一个无签名的证明以获得 SWA 签名、并且再循环传输这个有签名的证明的方式加入系统．证明中包括上面讨论过的身份信息．在证明被 SWA 密钥签署以后，证明被送往别的签名设备并附带一条指令要求把新设备加入到别的签名设备的内部表中．图 19 阐明了一个指令例子 281，包括命令 283 和证明 282．证明包括：新签署的设备身份码 285；签名设备的签名校验密钥证明 286（由生产厂商签署）；签名设备的加密密钥证明 289（也由生产厂商签署）．签名校验密钥和加密密钥也可以处在一个证明里．别的信息一定要在别的签名设备中循环，例如被新的签名设备使用的子密钥的身份码 291，和新设备附带的解密子密钥 292．一旦一个新设备加入了群里，它能：1、参与协议，产生新的主密钥并接收它的子密钥；2、作为一个后备单元以接收一个签名 SD 的内容；3、作为一个替换单元以接收一个修正后备签名设备所恢复的内容，这个签名设备要么被破坏了，要么被移出了服务．

图 20 阐明了移去一个签名设备的消息 293．消息 293 包括命令 295 和设备身份码 297．

复制密钥子密钥

由于多步签名过程的优越性，而且单个签名设备无法伪造一个签名或泄露足够伪造签名的信息，偷盗或破坏签名设备的危险性（后果）就被减小了．因此，例如当升级签名设备硬件或为了备份的目的时，签名设备的信息内容，包括 SWA 密钥子密钥，可以被发送到另一个设备．

密钥子密钥和其它信息的复制是通过提交由 SWA 密钥签署的一个请求来完成的，该请求是将某个特定的签名设备上的所有或一些信息复

制到第二台设备上．图 21a 描述了一个请求的例子，该请求要发送设备复制它的密钥子密钥．请求 301 最好包括：由 SWA 密钥签署的命令 303，识别设备的制造者 305（必须已包括在已认证的设备提供者的签名设备列表中）类型号 307（必须已包括在同意的类型列表中），以及序列号 309；用于接收设备的具有公开加密密钥的证明 311；被复制的密钥子密钥的 ID 码 313（或其它信息的指示）；以及发送设备 ID315．当签署的请求被相应的发送设备接收，发送设备将识别的密钥子密钥以及使用接收设备公开加密密钥的相关信息加密，然后发送设备将加密后的信息作为一条“增加密钥”消息输出给接收设备．图 21（b）描述了从发送设备发往接收设备的消息例子．请求 314 最好包括：由发送设备（-SD）签署的命令 316；接收设备 ID317；发送设备 ID318；加密密钥子密钥 319 的 ID 码；以及密钥子密钥拥有者 320 的 ID 码．接收部分指令也可以对于接收设备上的使用指定一个选定法定多数（或其它授权细节），但是，最好是接收到的密钥将根据接收设备的默认选定法定多数来使用．作为一个典型的操作过程，所有的系统操作员和有权用户将被告知复制已经完成，同时告知拥有备份的设备或存储介质的身份．

或者，该信息可以以加密的形式被复制到物理上安全（如，保存在暗室中）且脱机（不会遭受远程袭击）的存储设备中，以作备份．

改变选定请求

需要附加 SWA 密钥的签名设备的法定多数是由主设备在产生密钥子密钥时使用的系统设计参数．该法定多数可以通过重新合组密钥子密钥来恢复整个签名密钥，然后将该密钥分成数量增加了的密钥子密钥，这些子密钥按先前的密钥子密钥又被重新分布，但具有新的法定多数请求．

需要用来对特定的进行不完全签名的签名设备授权的授权代理的法定多数，可以不用重新初始化系统就进行改变．这样的改变最好通过向相应的签名设备提交由 SWA 密钥签署的请求来完成．或者，特定签名设备的授权代理可以通过提交一个只由本地授权代理签署的请求来改变本地的法定多数．需要改变选定子密钥的签名数目可以与用来授权签名设备时附加的 SWA 签名所需的相同，也可以不同．请注意，如果 SWA 密钥子密钥以加密的形式保存在签名设备中，且如果有权用户拥有解密密钥，如下文所述，则为了授权一个签名所需的法定多数，不应

该减少到少于对 SWA 密钥子密钥解密所需的子密钥的数目．在通常银行业的实际情况下，虽然一些有权用户可以在多个签名设备上有权限，但每台签名设备的有权者数目 N 不能小于 2．

加密存储密钥子密钥

如图 22 所示，在这种方案中，存储在一个签名设备 321 中的每个 SWA 密钥子密钥 323 是以加密后的形式 323 保存的．解密密钥（“KEY”)被分成各个子密钥，且每个授权代理的委托的的设备 325，327，329 各保存一个解密密钥．如上文所述，每一个对于签名设备附加不完全签名的请求，必须同时具有法定多数授权代理的签名．在该方案下，授权代理额外地向签名设备 321 发送解密密钥 331，333，335 的一个子密钥．签名设备则：

1）将解密密钥子密钥 337 组合起来，恢复解密密钥 347；

2）对 SWA 密钥的子密钥进行解密 339；

3）使用未加密 SWA 子密钥 341，将不完全签名 343 附加到文件上 345；

4）删除解密密钥 347；

5）删除解密密钥子密钥 331，333，335；以及

6）删除 342 未加密 SWA 密钥子密钥 341．

当发送文件到签名设备进行签名，授权代理把该代理的解密密钥子密钥包括在内，并签署该消息．在一般操作中，解密密钥子密钥是被保护的，因为所有网络上的通信是由接收者（即，当文件为了代理的签名在循环过程中的另一个授权代理，或当提交签名时的一个签名设备）的公开加密密钥加密的．或者，为了保护解密密钥子密钥，每个授权代理会为每条消息产生一个对话期密钥．（也就是，每次从一个授权代理传递一条包含密钥的消息到另一个授权代理或签名设备，一个新的对话期加密密钥被使用．）这样，全部的消息被对话期密钥加密．

用这种方法，明文的 SWA 密钥子密钥只是在它被用于附加不完全签名时才瞬时地存在．另外，解密密钥和解密密钥子密钥的全集也只是瞬时地存在．如果签名设备被窃，偷窃者最多只能恢复出 SWA 密钥子密钥的加密形式．

产生和分布加密密钥子密钥和解密密钥子密钥的过程见如下阐述，且在图 23 中描述．

1） 如上文中对基本方案的讨论，领导设备产生一个公开 SWA 认证密钥 351 和私有 SWA 签名密钥的子密钥 353， 355， 357．

2） 领导设备为每个 SWA 签名密钥的私有子密钥产生一个独立的公开/私有加密密钥对 359， 361（示出了一个 SWA 子密钥 357，可以理解，其它的子密钥将被类似地处理）．

3） 对每个私有加密密钥，领导设备采用 M 取 L 分割（L of M split）将私有解密密钥分为子密钥 363a,...363m，其中 M 为所有子密钥的总数，而 L 是重构私有解密密钥所需的最小的子密钥数． M 可以选成等于签名设备上的所有有权用户的总数，而 L 等于需要用各 SWA 密钥子密钥授权签名的授权代理的法定多数．

4） 领导主设备以相关的公开加密密钥 359 将每个 SWA 签名密钥 357 的子密钥加密，且将加密的 SWA 签名密钥的子密钥发送给一个独立的签名设备，并发送独立的私有解密密钥的 M 各子密钥．

5） 用于 SWA 密钥子密钥的私有解密密钥也可以被附带在（为安全起见，分布在）其它签名设备中，这样，任何私有解密密钥可以从签名设备恢复出来，但没有一个签名设备包含了恢复另一个设备的解密密钥所需的足够的信息．这样的任何特定签名设备的一般子密钥可以被发布，且取决于其他 SD 上的法定多数有权用户的同意．

6） 主设备从存储器中删除私有解密密钥、私有解密密钥子密钥、以及整个私有 SWA 签名密钥（如果它仍存在的话）．

当每个签名设备注册它的独立授权代理时，签名设备额外地发送一个解密密钥子密钥给每个授权代理，该子密钥由下述内容来识别： 1）用于解密密钥子密钥的识别号；和 2）用于相关 SWA 密钥子密钥的识别号．

例如，如果现有五个 SWA 签名密钥子密钥，（在签名时要有三个）且每个 SWA 密钥子密钥被不同的公开加密密钥加密，并且每个 SWA 密钥子密钥需要五个中的三个授权代理，那么每个解密密钥可以被分成五个子密钥，其中用任何三个都能恢复解密密钥．将会有二十五个解密密钥子密钥，每个签名设备分配五个给它的授权代理（用于它的自己的密钥）并拥有用于其它四个别的设备的每一个的解密密钥的一个子密钥．

通过这种方法，需要对附加不完全签名的签名设备进行授权的法定

多数的授权代理，将也有足够数目的解密密钥子密钥，使得签名设备能够对用于每个签名操作的 SWA 密钥子密钥瞬时地解密.

如果一个或多个授权代理丢失了它们的密钥（例如，丢失了它们的委托的设备的智能卡），那么新的智能卡将被注册到同一签名设备上。解密密钥子密钥可以从其它签名设备上恢复出来，并且 可以重新装到新注册的智能卡上，方法是提交由 SWA 签名密钥签署的电子消息，使签名设备将解密密钥子密钥转送到新注册的签名设备上. 另一种方法是，在 SWA 允许的前提下，一个特定的设备可以接收所有类型的子密钥，对它的签名子密钥解密，生成一个新的加密密钥对，在公开密钥下重新加密签名子密钥，将新的私有解密密钥分成新的子密钥，并将这些子密钥重新分布到有关有权用户的委托的的设备上，同时注意到要用那些接收有权用户的委托设备的公开密钥将上述子密钥加密.

作为另一种备份方法，解密密钥子密钥可以脱机地附带在独立的委托设备中，如同时申请的美国专利申请 Nos.08/181,859 和 08/277，438 中所述.

密码的心跳

作为进一步的保护措施，每个签名设备接收一个周期性的数据输入（“心跳”），如果它中断了，签名设备置就变为非活动状态. 该心跳将从不同于签名设备的位置产生，这样，如果偷窃者企图偷窃一个签名设备，他们也必须进入一个独立的房间或密室去得到心跳源. 如果它们得不到心跳源，签名设备将会被去激活，成为无用.

在一种实现方案中，每个签名设备提供一个加密密钥给心跳源. 心跳源周期性地向签名设备发送加密了的消息. 如果签名设备在一个时间段内没有从心跳源接收到最小数量的消息，则签名设备将删除它的内部存储器，或采取其它的规避行动. 消息可以是空消息，或简单的消息，这些消息必须由心跳源用 SD 赋予它的公开的有规律的密钥来加密. 或者，该消息是一个伪随机字符串，由伪随机数产生器（ RNG ）在心跳源中生成，并由签名设备中的同步的（ RNG ）确认.

可以设立多个心跳源，使得一个签名设备必须在一个时间周期内从至少一个（或最小数量的）心跳源接收到消息. 如果一个心跳源由于设备故障或电源掉电而脱机，它不会引发签名设备存储器过早的删除. 心跳通信中使用的密钥会被备份在多个位置的子密钥中.

在第二种实现方案中，每个签名设备会发送一个查询到网络上的相关一组（“卫星”）设备中，且仅在至少法定多数的相关设备响应后，才继续操作．在意外掉电期和通信检修间，得到法定多数的响应后，才允许操作继续．

使用卫星设备虽然更加复杂，但增加了物理上的安全性，它可以被应用在低安全的环境中，而不用使用密室、卫兵、摄象机等来提高安全设施．

签名设备和它的心跳源或卫星设备之间的通信链路可以是公共网．如果一个签名设备被报失，它的相关卫星单元可以被系统操作员停止活动，以防止偷窃者接上通信链路改变路由而将心跳引导到被窃设备．

例如，签名设备可以在美国，而它的相关卫星设备在欧洲．当签名设备被偷窃时，欧洲的卫星设备被操作员卸下线路．欧洲代理因错误动作的责任将最少，因为卫星设备的卸下只短时间地影响新的签名操作．先前签署的签名仍旧有效．或者可在签名设备和它的卫星设备或心跳源之间提供安全的物理线路以替代公共网．

产生额外的主控密钥

用一个 SWA 密钥建立了安全、多步签名系统之后，产生一些用于其它目的的额外的主控密钥则相对简单。 SWA 签名密钥控制系统管理，而主控密钥可以代表其它的合法实体，用来签署其它认证消息或文件．产生和管理其它主控密钥的方法与 SWA 密钥类似，但没有中间的临时认证步骤．该方法进行如下：

1） 指定一个签名设备为“领导”设备（它不必和产生 SWA 签名密钥的“领导”设备相同）．

2） 输入一个用于接收主控密钥子密钥的签名设备的公开密钥证明列表．

3） 为该主控密钥输入一个识别码和一个逻辑名．

4） 在签名设备中建立安全通信通道（最好使用每个相关签名设备的加密密钥证明）．

5） 可选地从每个签名设备获得随机资料．

6） 产生一个新的“主控”公开/私有密钥对．

7） 分布私有密钥子密钥（可选地为每个子密钥加密，并分布解密

密钥的子密钥）。

8） 删除整个主控私有密钥（如果它被保存），并由主签名设备删除所有的不保留的子密钥.

该过程也可以被用来代替 SWA 签名，方法是向每个签名设备发送一个额外的、由（旧的） SWA 签名密钥签署的命令，将新的主控密钥当作 SWA 签名密钥来安装. 通常，主控密钥与 SWA 密钥有不同的用途，且许多主控密钥的子密钥可以共存在签名设备中. 先前产生的主控密钥（除了 SWA 签名密钥）可以通过提交一个由 SWA 签名密钥签署的删除主控密钥段的消息而从系统中被删除.

文件和签名寻迹

为了支持系统中的文件流的管理，最好给每个要签署的文件分配一个唯一的识别码. 以下的信息可以被包含在每个文件的标题里，以供消息服务者或授权者使用:

1） 用来签署文件的密钥的签名密钥识别码.

2） 为完成签名和/或已提供的不完全签名所需要的所有的不完全签名数.

3） 已经被用来签名的密钥段识别码.

4） 已经签名的签名设备的身份（例如，逻辑设备名）.

签名设备的连锁环

如上文所述，一个使用多步签名系统的根 CA，一般将认证位于其它商业和政府机构中的从属的 CA. 可以这样假定，一个大的金融中心银行可能认证一个州政府的主要代理. 该州代理接下来可以认证一个公司. 这样就以一种使现有的政治、经济和社会组织所遵从的方式灵活地分布认证过程.

但是，每个中间级的 CA 必须在它的签名密钥上保持可靠的安全性. 除了银行、一些大的公司和一些政府机构，很少有些机构按传统保持的多个高安全数据处理设施和存储用的密室的. 例如，一个中间级的 CA 会拥有至少一个名义上的安全物理位置，比如数据中心或密室操作，但是缺乏资金去提供上文所述的多设备机制所需的多个地点. 另一种情况是，中间级 CA 可能没有一个真正安全的位置.

然而，低安全性的中间级 CA（诸如一个团体 CA）可以建立它们自己的签名环（如上文中所述），并将这些中间级环连锁到父 CA（诸

如银行或安全政府代理）的更高安全性的环上．当把（1）密钥的持有和正式的控制，（2）管理和备份职责，以及（3）物理地具有设备分开时就可以这样做．

连锁环结构可以如图24所示产生，方法是使一个中间级CA371在它自己的安全地点维护一个或多个中间级签名设备373，375，377．额外的中间级签名设备379，381将在父CA383的安全地点维护，且甚至包括构成父（根）CA环383的一些或所有的相同设备379，381（因此叫“连锁环”）．父CA可以维护一些签名设备385，387，389，它们独立于任何特定的中间级CA383．上文中阐述的签名设备不需要额外的修改以具有额外的主控密钥，每个主控密钥分别被不同的授权代理391a，391b所拥有和控制，每个代理具有以不同方式组合的主控密钥．

中间级CA用它自身的签名设备之一作为“领导”设备，以便如上文所述开始密钥的产生和子密钥分布协议，并授权自己的职员作为授权代理391b．新CA主控密钥的一些子密钥会保存在它自己的签名设备373，375，377中，而其它的将保存在它的父CA379，381中．发布签名的权利可以只保留在密钥持有者的职员手中，虽然在紧急情况下，他们也可以代表一些赋予父CA机构的职员的权利．

此后，中间级CA基于他们职员所拥有的智能卡产生的签名，开始CA签名的中间步骤的签署，并将那些请求引导到它自己的签名设备，和/或引导到父CA所拥有的设备．实际上，签名设备不需要与父CA同处一处，可以位于其它具有安全地点并可以通信访问的的CA的位置．

完全的租用业务

一个连一个安全设施都没有的机构，可能仍旧希望能产生证明并仍旧可以成为一个CA．该机构可以租用已经由不同的银行或其它CA建立起来的在安全位置的签名设备．该机构拥有用于它的授权代理的智能卡，并将签名请求通过通信网络引导到签名设备．这样，产生密钥、发布签名、以及执行其它管理任务的处理过程，依照与设备所有者约定的确认办法，可以在本地银行的物理控制下在设备中进行．

该机构的职员可以到本地安全（银行）设施去目证密钥产生协议，用该协议，由此，他们的新的签名密钥被产生、分开、以及分布到一些他们选定的主机设施（可能是其它的银行或同一个银行的其它位置）的

每一个上．同时，他们也可以根据需要，分配适当的管理备份权利．

该机构然后可以发布正式的签名和证明，而不需要建立他们自己的安全数据中心或密室设施，同时仍旧实际上获得上述系统所有的安全收益．

签名代表

当一个授权代理短时性地不可使用（由于在假期，能力不够，等），就需要某种形式的签名权代表．不希望人类操作员将他/她的智能卡（以及相应的个人识别号或密钥）借给其它人，因为这将产生不可管理的安全方面的危险性．

一种可选的代表机制是，一个原有的授权代理（“主用户”）向替代的授权代理（“代表”）发布一个特殊的“代表”证明．该证明由主要用户签署并将识别代表以及代表的公开签名认证密钥．该代表证明还会包括一个时间期限，在该时间期限中代表证明（因而具有代表权利）是有效的．（参看 Sudia 和 Ankeney,“数字签名的商业化”，1993．）使用他/她个人智能卡的代表将用代表本人的签名签署文件，并将代表证明附上．最终的文件将由代表而不是主用户签署，而文件的接收方必须经过额外的认证代表签名和代表证明的步骤．这种做法，子密钥部分地依赖于系统的所有公共用户都具有这种认证能力，并且如果所述的权利必须在过期之前被删除，还依赖于用户能够对解除信息（或“热点列表”）的源能进行良好的访问．

一个优选的方法是允许代表以一种安全的方式使用主用户的智能卡，它实际上通过主用户的智能卡为人类的主用户去替代人类的代表．所以，代表可以使用主用户的智能卡来附加主用户的签名，整个文件的接收者就不用承担额外的负担来认证和评估另一个复杂的证明．

当主用户希望委派签名权，该主用户发布一个“替代”证明 409 给代表，如图 25 所示．该替代证明标识主用户 ID 411、代表 ID 413、用于使主智能卡识别代理的装置（一般是代表的公开认证密钥 417）、以及时间期限 415，在该期限中替代证明是有效的．主用户可以识别多个个体，任何一个个体可以对智能卡授权，或者这些个体中的多个个体的组，它们一同对智能卡授权．这些方法的前面的内容在以下的 Addison fischer 的专利(美国专利 Nos.4,868,877, 5,005,200,以及专利 5,214,702)中阐述．

如图 25 所示，当一个代表想代表主用户签署文件的时候，准备并以特定的形式签署一个与主用户的智能卡 407 通信的请求 405．附加或包含在该消息内的是替代证明 409．如果多个代表需要对主用户的智能卡授权，他们可以以上文所述的类似于多个授权代理签署提交给签名设备的请求的方式顺序地签署请求．当接收签名请求时，主用户的智能卡将确认提交请求的用户的签名与替代证明中最初特定的公开密钥相符，加上主用户的签名 419，并将签名后的文件以通常的方式转送到签名设备 421（或其它的目的地）．

主用户的智能卡 407 会可以用实物交给代表．代表权利时限的存在提供了一个“时间锁”，这样代表只能在时限期间使用主用户的智能卡．如上文所述，主用户的权利也被限定在一个固定的时间段内．这些限制减少了被偷窃的后果，并因此允许主用户和代表能将主用户的智能卡保存在一个相对低安全性的办公室环境中．当时限到期后，智能卡对于任何猜测密钥的攻击不再是脆弱的．（实际上，它将不受攻击的影响，即使当主要用户或代表将他/她的个人识别号直接写到卡上．）

将智能卡放置到密室或其它加锁的环境中，并将卡插入读卡器，在其中卡能够被电子方式读取，而不能物理地读取，通过这种额外的保护，可以防止丢失或物理上的攻击．这种方式下，所有的上述动作都可以执行，但是没有人能够物理地拥有智能卡．

例如，一个主用户可能是一个负责购买的副总裁，他希望在他外出商议一个未决的交易时，用他的秘书来代表他的特定签名权．替代证明应该明确他的智能卡只有在接收到由以下条件签署的签名请求时才能代替副总裁的签名：（a）由替代证明中所指定的秘书；（b）由采购部的其它有主要签名权的人同签．该副总裁将他的卡放在密室中的读卡器上，然后离开．

为了获得副总裁的签名，秘书可以准备好要签署的文件，并用她的桌面计算机终端计算相关散列，附加上最终的接收者会需要的副总裁的公开密钥证明，然后将它们在一个消息中发送到另一个采购代理处．该采购代理其同签署相同的散列，并附上他的公开密钥证明以及他的授权证明，该证明保证了他的采购权．另一个采购代理将它们在一个消息中经过一个局域网发送到副总裁的智能卡．如果副总裁的卡也包含了确认过的认证单位的公开密钥的拷贝，这些单位产生了诸如 SWA 的这些证

明，那么副总裁的卡便决定签名和证明都是有效的，并将副总裁的签名加在文件上。该卡可能还要求所有的这些证明与最近签署的 CRL 的或从本地认定的 CRL 身份良好的证明包含在一起.

这种代表机制的优越性表现在可以对主用户的智能卡重新编程的能力．主用户的智能卡是已委托设备，它具有已知的安全特性，其中之一必须是参加新指令（例如，替代证明）的安全下载的能力，例如在共同待决的美国专利申请 08/181， 859 和 08/272， 203（ Sudia 密钥契约 parent 和 Sudia 密钥契约 CIP ）中所述.

前述的代表机制可以被一般化，这样许多高价值的最终用户数字签名密钥实际上在防篡改安全模块（ TRSM ）下产生和使用，该模块被保存在安全密室或数据中心里，而所述签名的授权来自由认可用户签署的签名请求消息，被认可用户则被给予非正式的（时间锁定）的由他们携带的智能卡．这些 TRSM 将保持抗篡改安全性，防止任何数据中心人员读取用户私有密钥，但是可以被设计成能包含许多不同用户的密钥，每个用户可基于一些单个非正式签名、或一些预定的签名和授权的组合而被授权操作。

除了简单的用户短期离开或缺席时的代表，代表机制的另一种用途是一个这样的系统或方法，其中所述的可编程签名请求将被作为一个卡（或一个包含在公用 TRSM 中的密钥）来完成商业或公司环境中的“主席”或其它角色的签名.

了解上过实施例之后，本技术领域的技术人员将可以在本发明核心和范围内做不同变化．上述的实施方案是作为示例，并不过度限定发明的范围，发明范围由以下权利要求定义.

说 明 书 附 图

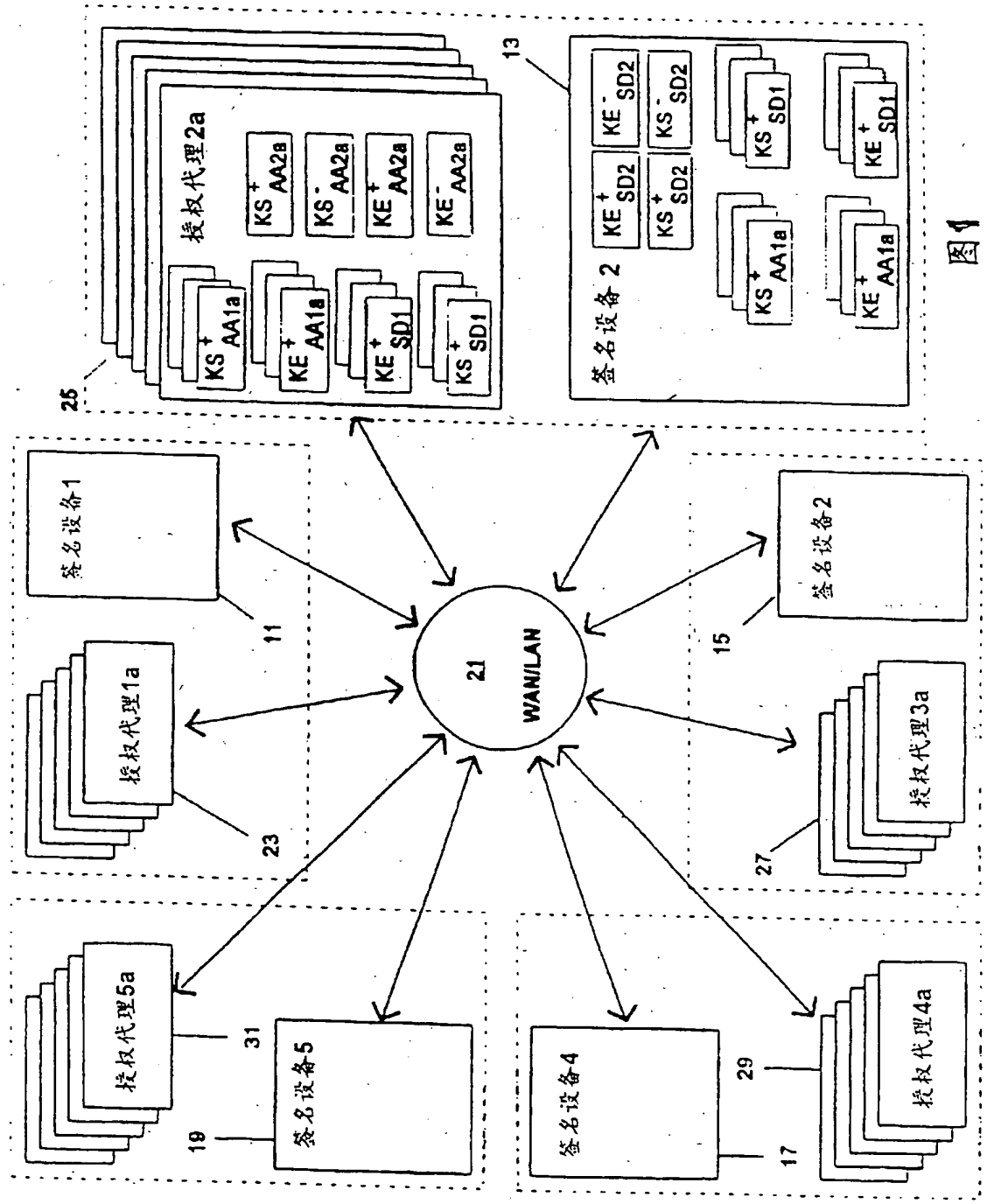

授权代理2a
KS+ AA2a
KS- AA2a
KE+ AA2a
KE- AA2a
KS+ AA1a
KE+ AA1a
KE+ SD1
KS+ SD1
25
签名设备 2
KE+ SD2
KS+ SD2
KE- SD2
KS- SD2
KS+ AA1a
KE+ AA1a
KS+ SD1
KE+ SD1
13
签名设备1
授权代理1a
11
23
21
WAN/LAN
签名设备2
授权代理3a
15
27
授权代理5a
签名设备5
31
19
签名设备4
授权代理4a
29
17

图1

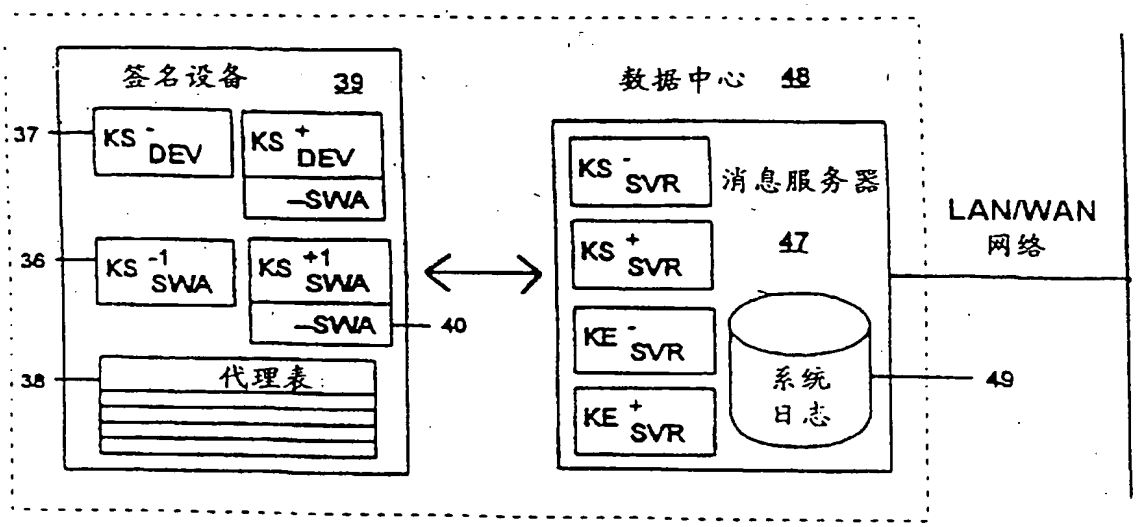

签名设备
39
37
KS - DEV
KS + DEV
–SWA
36
KS -1 SWA
KS +1 SWA
–SWA
40
38
代理表
数据中心
48
KS - SVR
消息服务器
KS + SVR
47
KE - SVR
KE + SVR
系统日志
49
LAN/WAN
网络

图 2

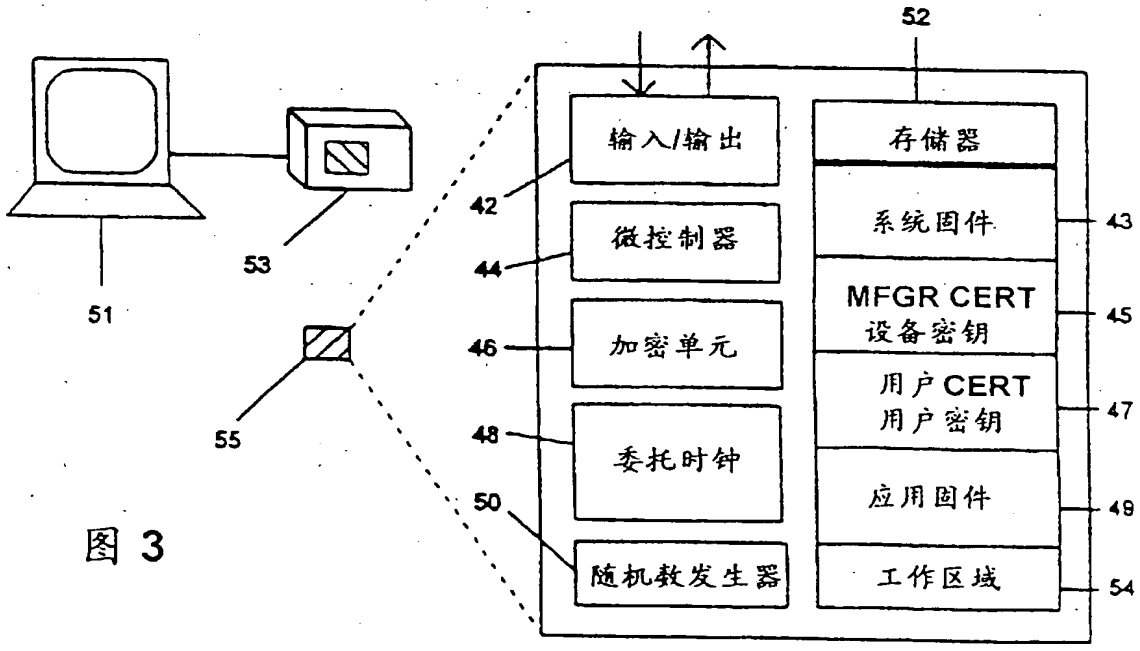

52
输入/输出
存储器
42
微控制器
系统固件
43
44
MFGR CERT
设备密钥
45
46
加密单元
用户 CERT
用户密钥
47
48
委托时钟
应用固件
49
50
随机数发生器
工作区域
54
51
53
55

图 3

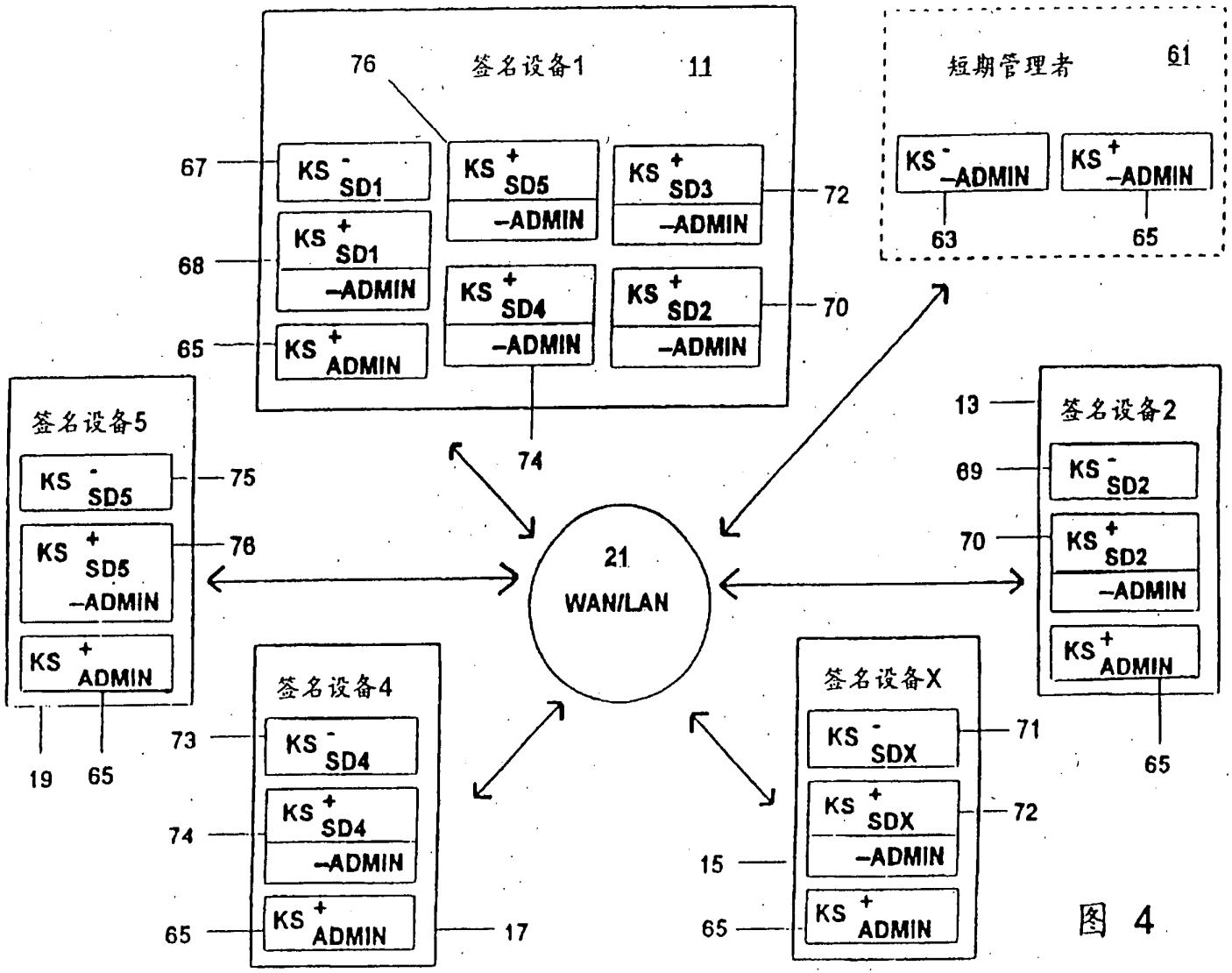

76
签名设备1
11
短期管理者
61
67
68
65
72
70
74
63
65
签名设备5
75
76
19
65
13
签名设备2
69
70
65
21
WAN/LAN
签名设备4
73
74
65
17
签名设备X
71
72
15
65

图 4

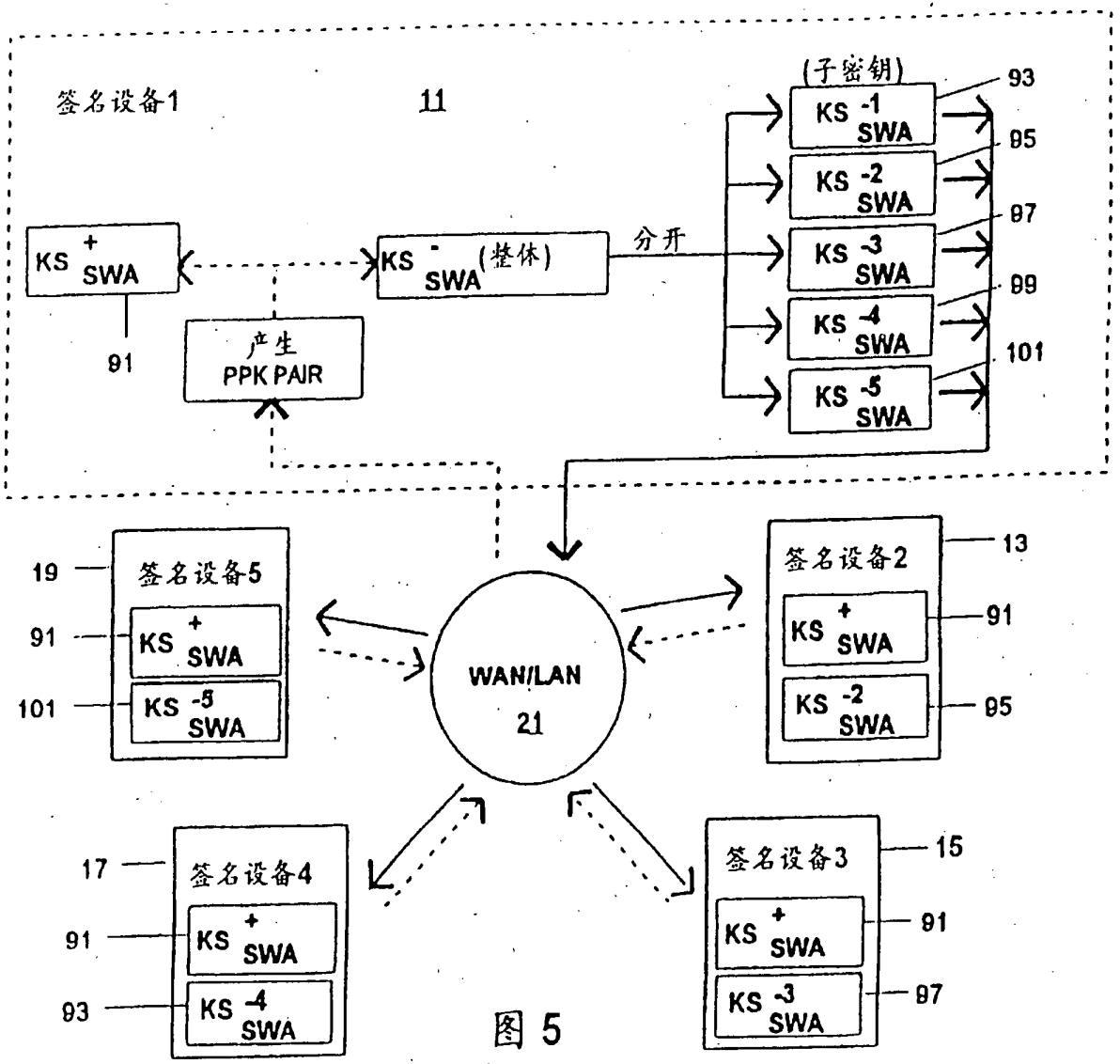

签名设备1
11
(子密钥)
KS -1 SWA
KS -2 SWA
KS -3 SWA
KS -4 SWA
KS -5 SWA
93
95
97
99
101
KS + SWA
KS - SWA (整体)
分开
91
产生 PPK PAIR
WAN/LAN 21
签名设备5
19
91
101
KS + SWA
KS -5 SWA
签名设备2
13
91
95
KS + SWA
KS -2 SWA
签名设备4
17
91
93
KS + SWA
KS -4 SWA
签名设备3
15
91
97
KS + SWA
KS -3 SWA

图 5

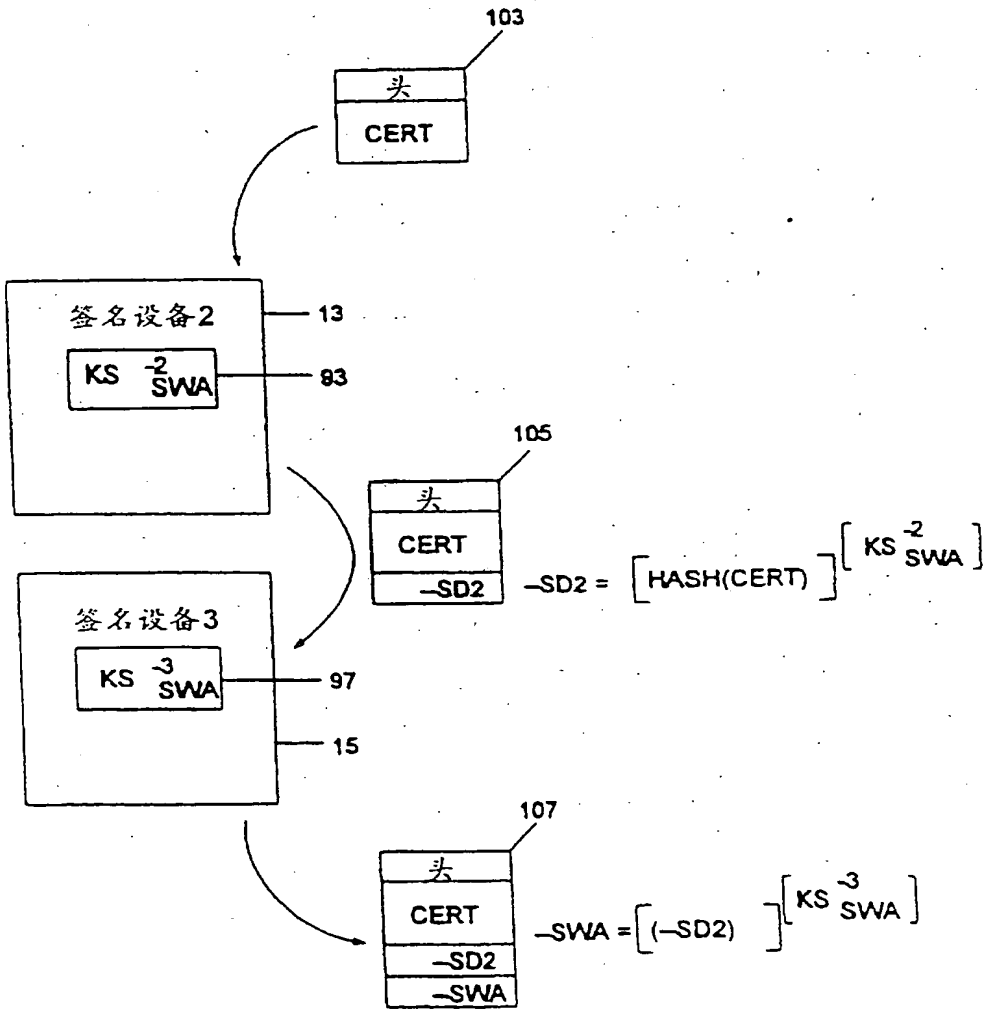

103
头
CERT
签名设备2
13
KS -2 SWA
83
105
头
CERT
–SD2
–SD2 = [HASH(CERT)] [KS -2 SWA]
签名设备3
KS -3 SWA
97
15
107
头
CERT
–SD2
–SWA
–SWA = [(–SD2)] [KS -3 SWA]

图 6

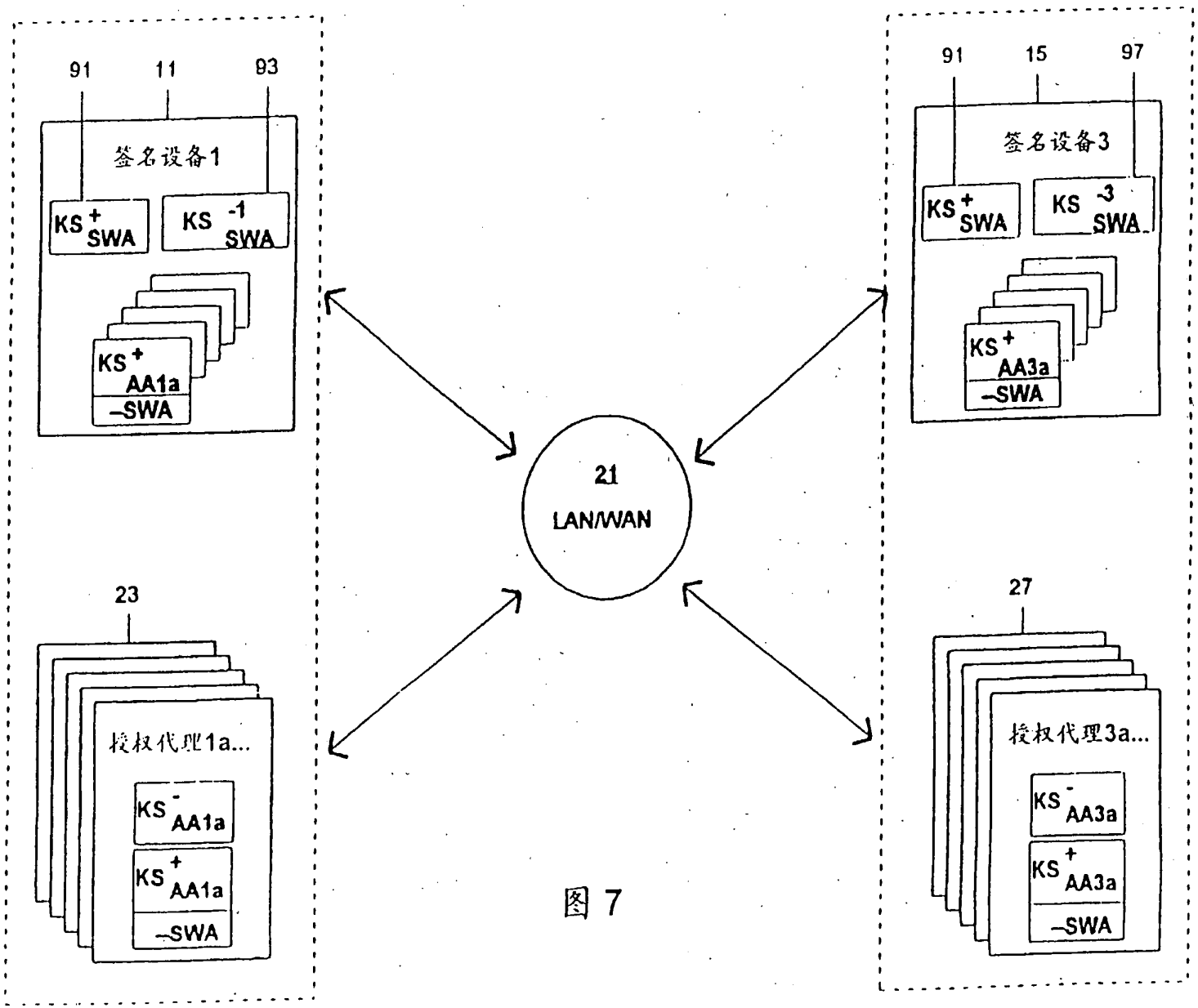

91
11
93
签名设备1
KS +SWA
KS -1SWA
KS +AA1a
–SWA
91
15
97
签名设备3
KS +SWA
KS -3SWA
KS +AA3a
–SWA
21
LAN/WAN
23
授权代理1a...
KS -AA1a
KS +AA1a
–SWA
27
授权代理3a...
KS -AA3a
KS +AA3a
–SWA

图 7

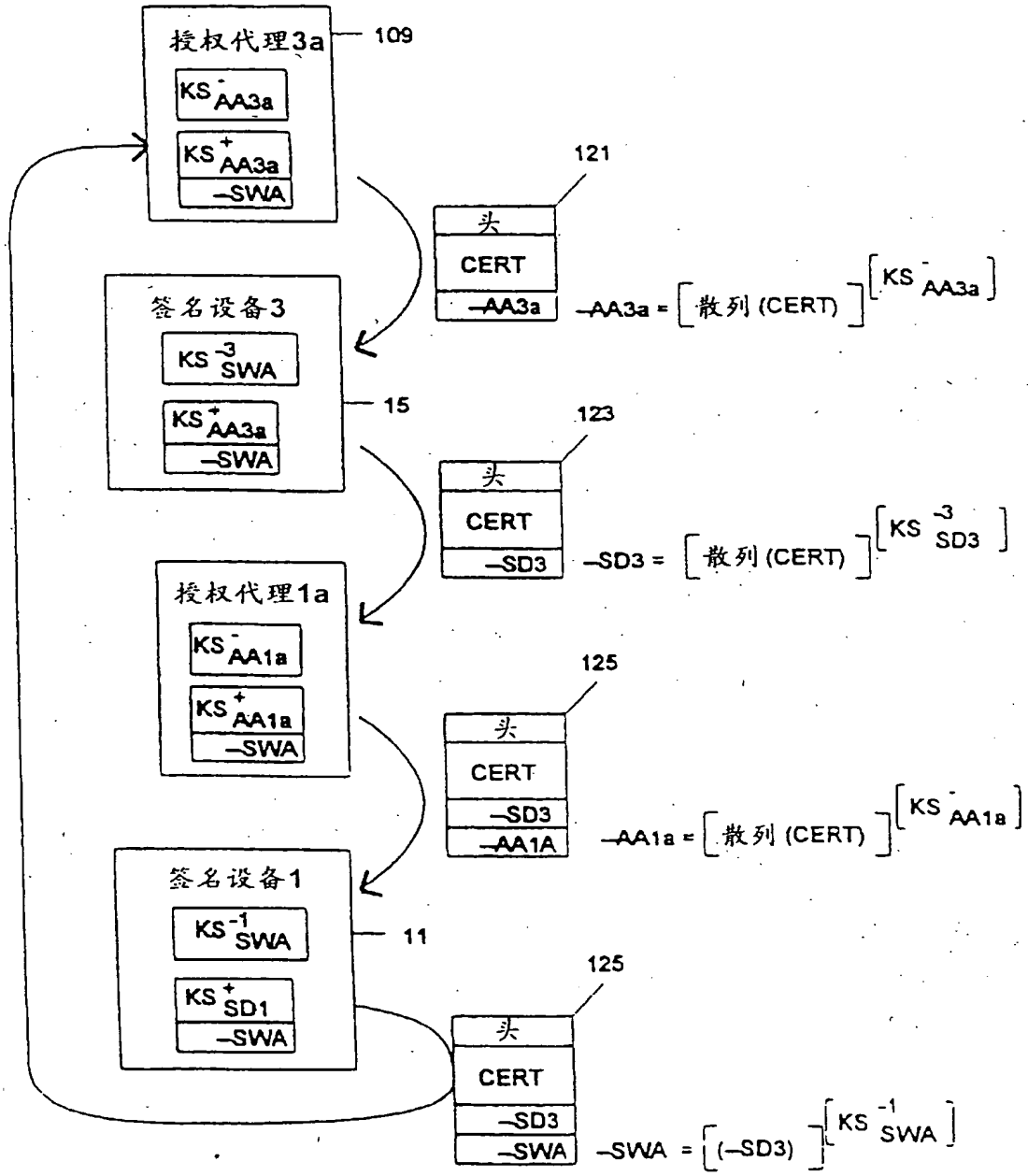

授权代理3a
109
KS^{-}_{AA3a}
KS^{+}_{AA3a}
–SWA
121
头
CERT
–AA3a
–AA3a = [散列 (CERT)] $[KS^{-}_{AA3a}]$
签名设备3
KS^{-3}_{SWA}
KS^{+}_{AA3a}
–SWA
15
123
头
CERT
–SD3
–SD3 = [散列 (CERT)] $[KS^{-3}_{SD3}]$
授权代理1a
KS^{-}_{AA1a}
KS^{+}_{AA1a}
–SWA
125
头
CERT
–SD3
–AA1A
–AA1a = [散列 (CERT)] $[KS^{-}_{AA1a}]$
签名设备1
KS^{-1}_{SWA}
KS^{+}_{SD1}
–SWA
11
125
头
CERT
–SD3
–SWA
–SWA = [(–SD3)] $[KS^{-1}_{SWA}]$

图 8

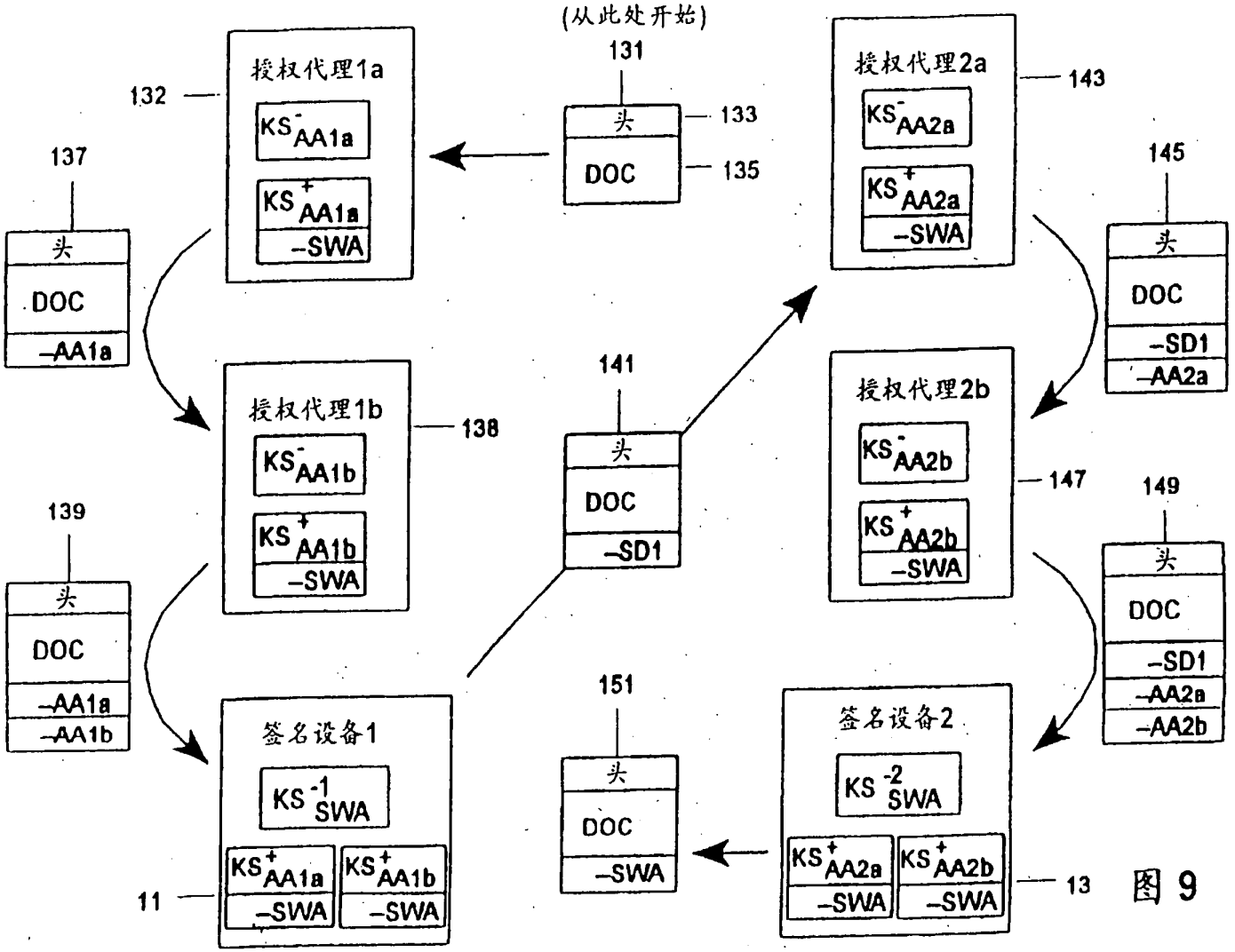

(从此处开始)
131
头
DOC
133
135
授权代理1a
132
KS-AA1a
KS+AA1a
-SWA
137
头
DOC
-AA1a
授权代理1b
138
KS-AA1b
KS+AA1b
-SWA
139
头
DOC
-AA1a
-AA1b
签名设备1
KS-1SWA
KS+AA1a
-SWA
KS+AA1b
-SWA
11
141
头
DOC
-SD1
授权代理2a
143
KS-AA2a
KS+AA2a
-SWA
145
头
DOC
-SD1
-AA2a
授权代理2b
147
KS-AA2b
KS+AA2b
-SWA
149
头
DOC
-SD1
-AA2a
-AA2b
151
头
DOC
-SWA
签名设备2
KS-2SWA
KS+AA2a
-SWA
KS+AA2b
-SWA
13

图 9

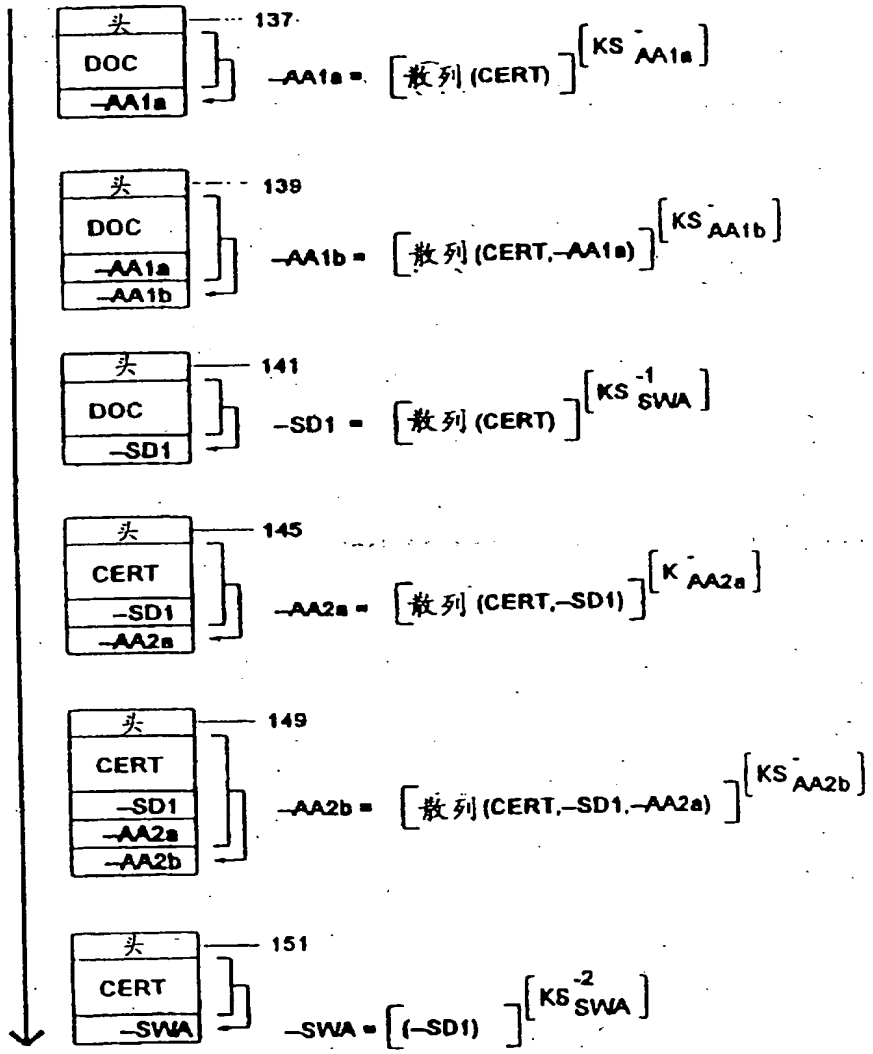

头
DOC
–AA1a
137
–AA1a = [散列(CERT)] [KS -1 AA1a]
头
DOC
–AA1a
–AA1b
139
–AA1b = [散列(CERT,–AA1a)] [KS -1 AA1b]
头
DOC
–SD1
141
–SD1 = [散列(CERT)] [KS -1 SWA]
头
CERT
–SD1
–AA2a
145
–AA2a = [散列(CERT,–SD1)] [K -1 AA2a]
头
CERT
–SD1
–AA2a
–AA2b
149
–AA2b = [散列(CERT,–SD1,–AA2a)] [KS -1 AA2b]
头
CERT
–SWA
151
–SWA = [(–SD1)] [KS -2 SWA]

图 10

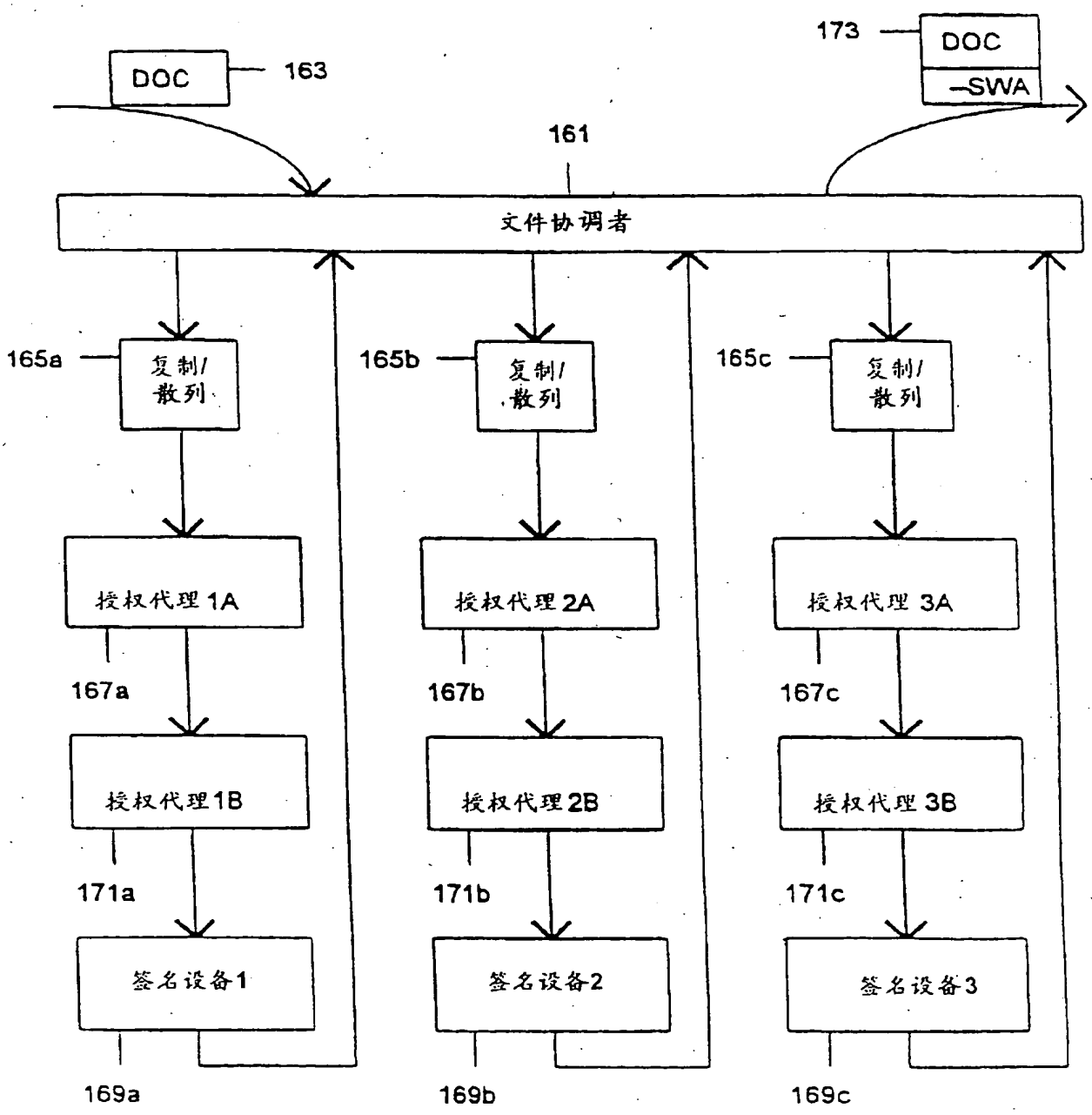

DOC
163
173
DOC
SWA
161
文件协调者
165a
复制/
散列
165b
复制/
散列
165c
复制/
散列
授权代理 1A
授权代理 2A
授权代理 3A
167a
167b
167c
授权代理1B
授权代理2B
授权代理 3B
171a
171b
171c
签名设备1
签名设备2
签名设备3
169a
169b
169c

图 11

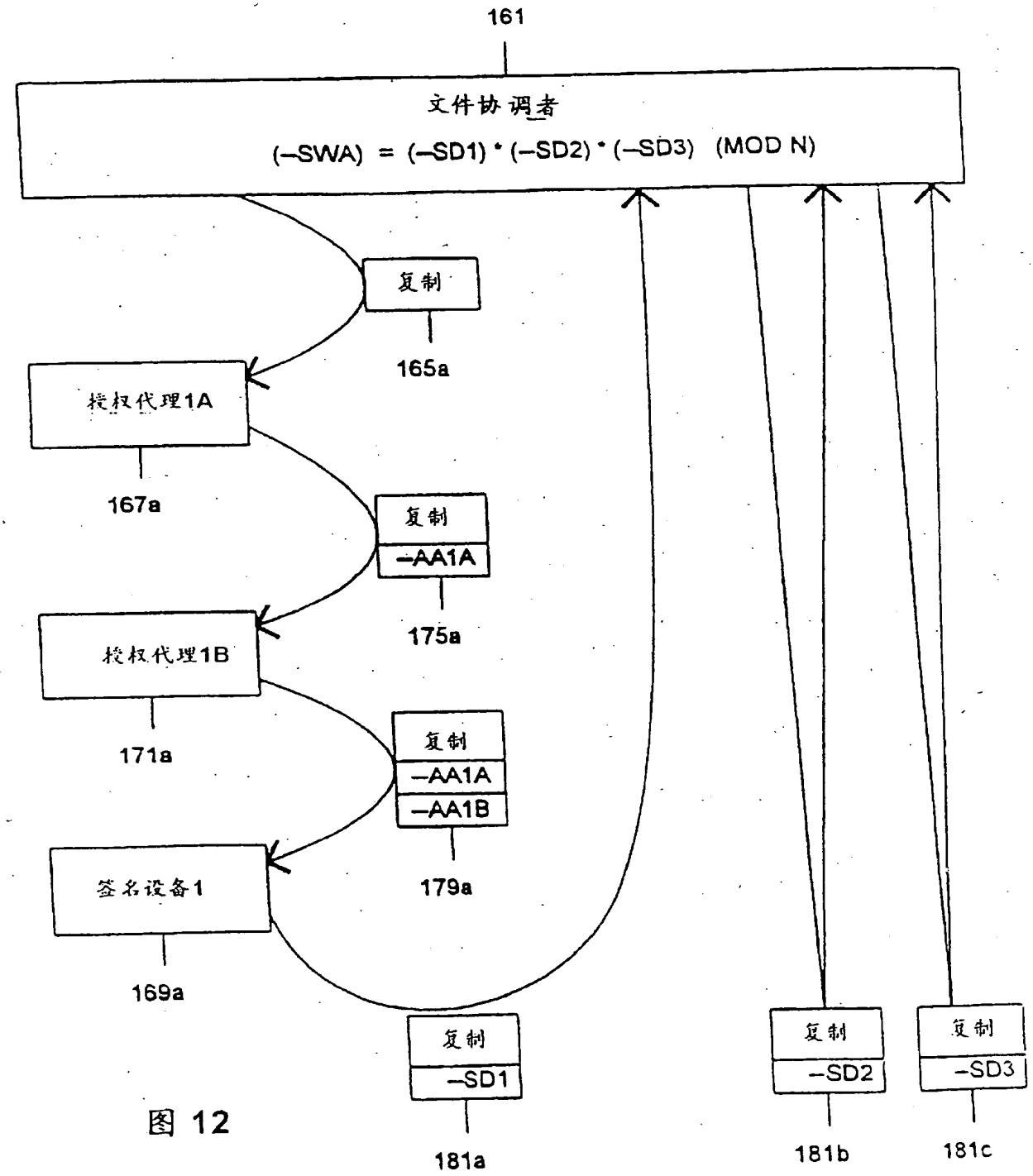

161
文件协调者
(–SWA) = (–SD1) * (–SD2) * (–SD3) (MOD N)
复制
165a
授权代理1A
167a
复制
–AA1A
175a
授权代理1B
171a
复制
–AA1A
–AA1B
179a
签名设备1
169a
复制
–SD1
181a
复制
–SD2
181b
复制
–SD3
181c

图 12

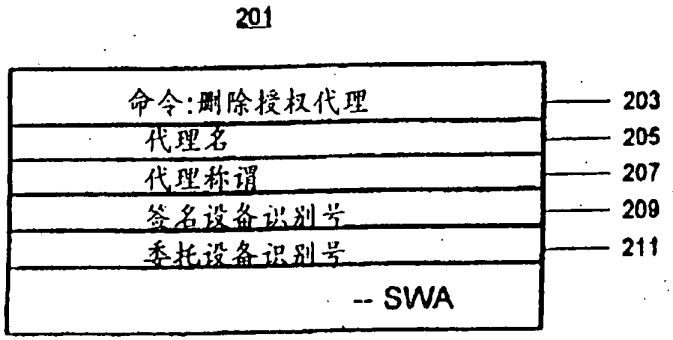

201
命令:删除授权代理
代理名
代理称谓
签名设备识别号
委托设备识别号
-- SWA
203
205
207
209
211

图 13

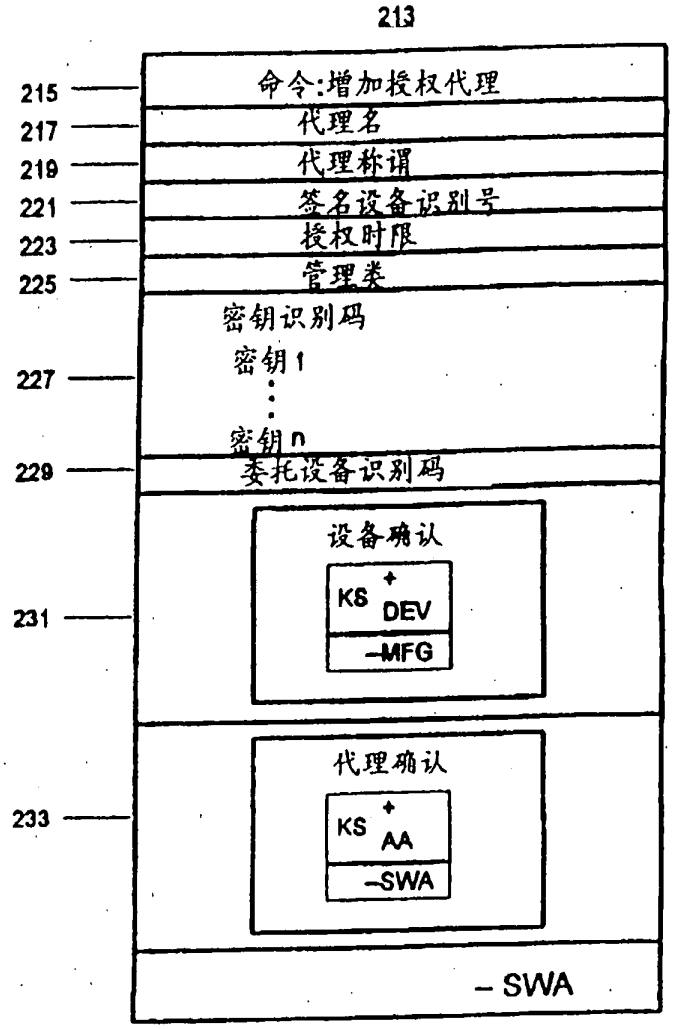

213
命令:增加授权代理
代理名
代理称谓
签名设备识别号
授权时限
管理类
密钥识别码
密钥 1
密钥 n
委托设备识别码
设备确认
KS + DEV
-MFG
代理确认
KS + AA
-SWA
- SWA
215
217
219
221
223
225
227
229
231
233

图 14

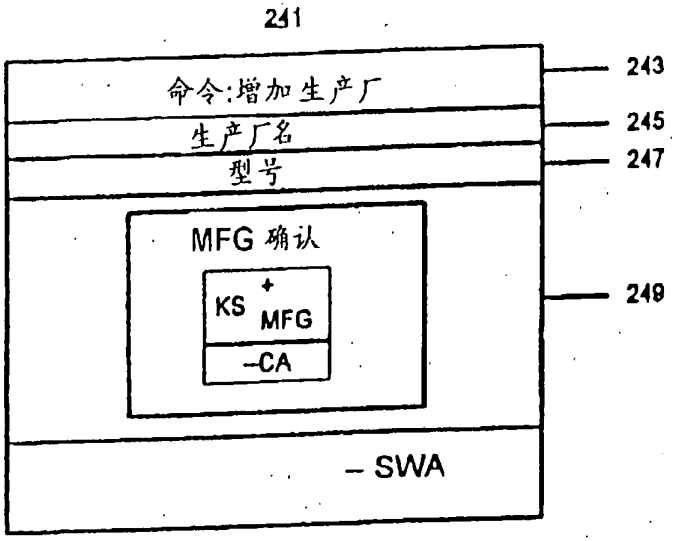


图 15

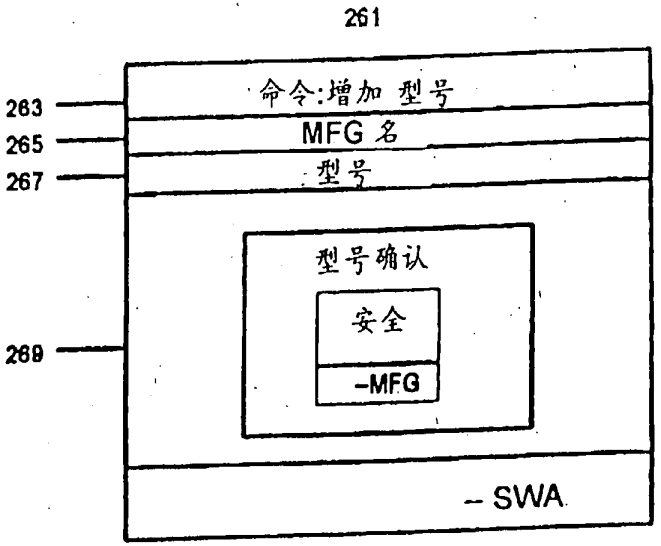


图 17

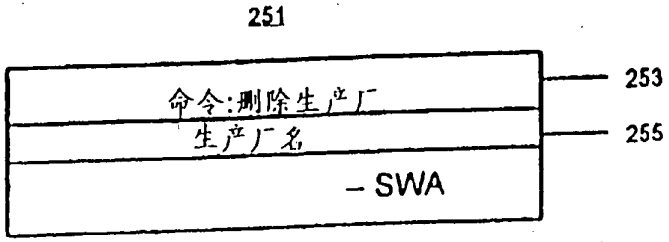


图 16

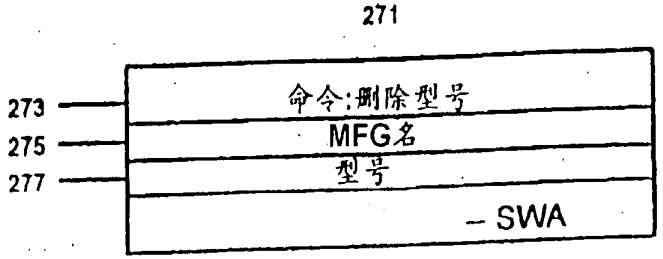


图 18

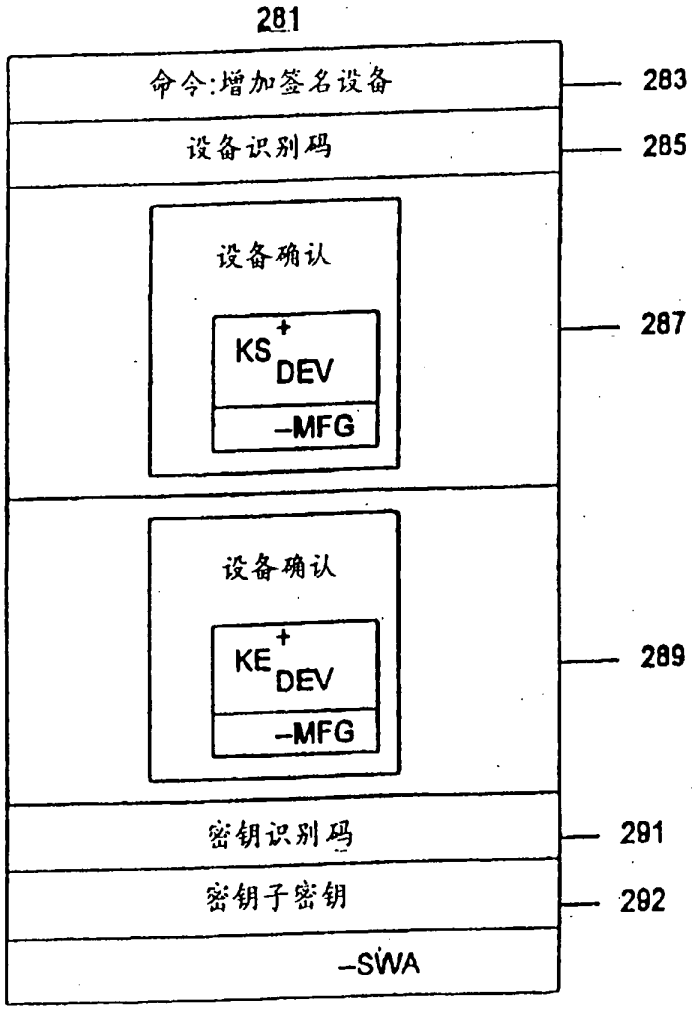

281
命令:增加签名设备
283
设备识别码
285
设备确认
KS+DEV
-MFG
287
设备确认
KE+DEV
-MFG
289
密钥识别码
291
密钥子密钥
292
-SWA

图 19

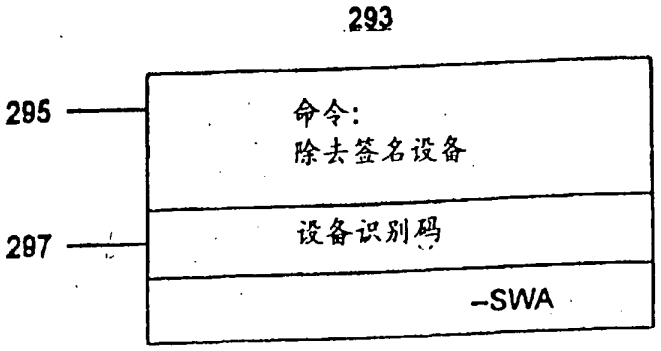

293
295
命令:
除去签名设备
297
设备识别码
-SWA

图 20

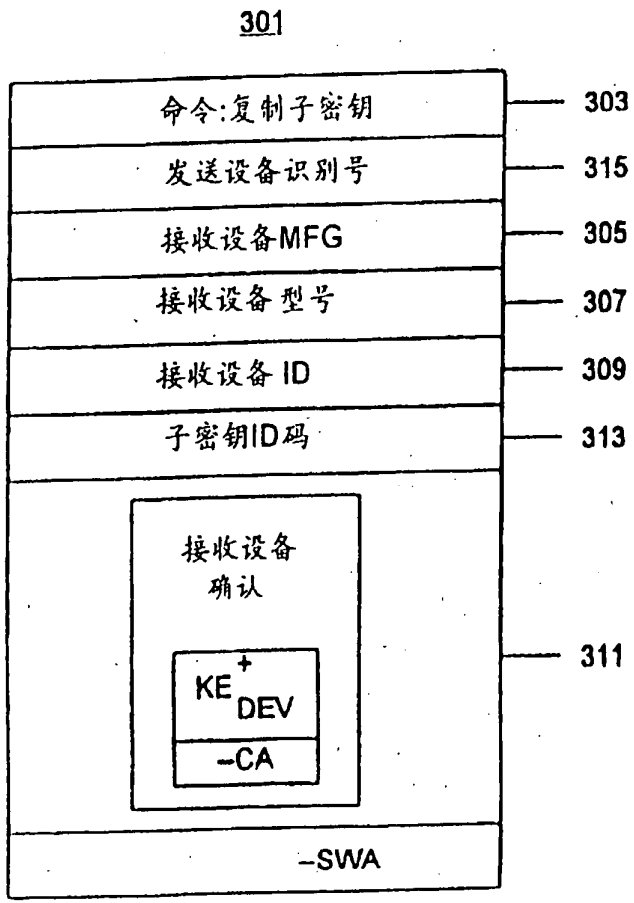

301
命令:复制子密钥
303
发送设备识别号
315
接收设备MFG
305
接收设备型号
307
接收设备ID
309
子密钥ID码
313
接收设备
确认
KE+DEV
-CA
311
-SWA

图 21 a

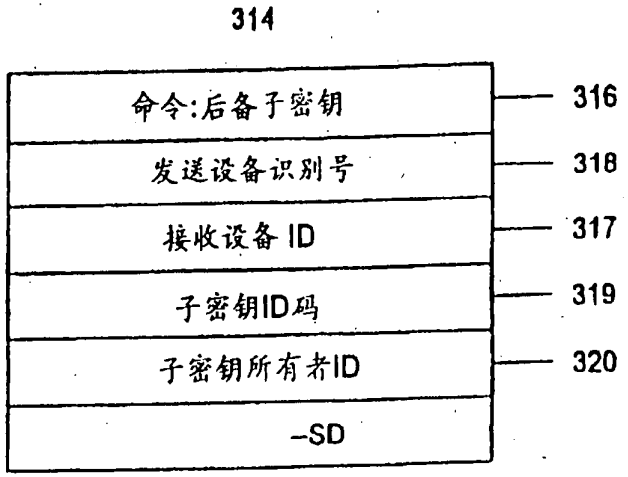

314
命令:后备子密钥
316
发送设备识别号
318
接收设备ID
317
子密钥ID码
319
子密钥所有者ID
320
-SD

图 21 b

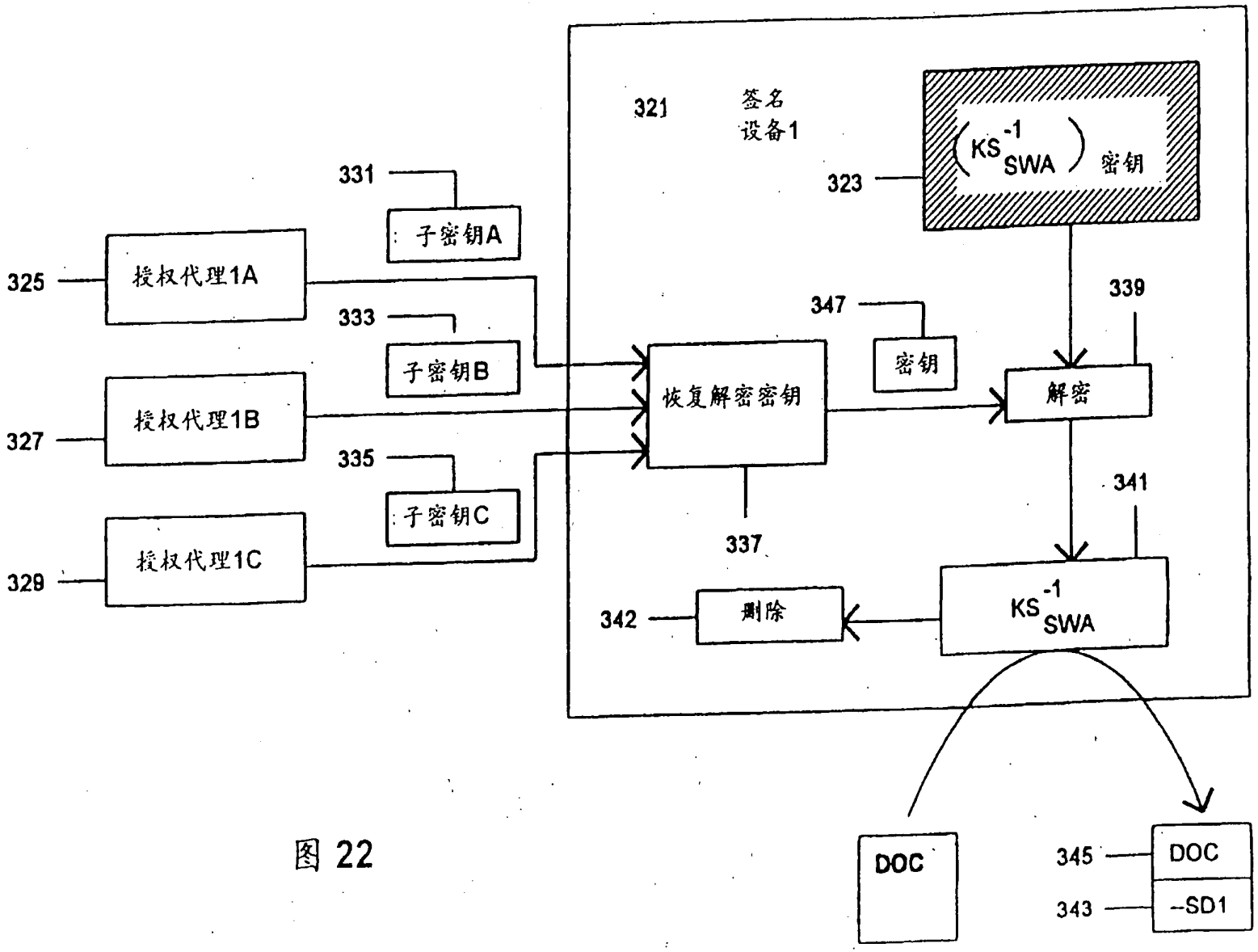

321
签名
设备1
323
KS^{-1}_{SWA} 密钥
331
子密钥A
325
授权代理1A
333
子密钥B
327
授权代理1B
335
子密钥C
329
授权代理1C
恢复解密密钥
347
密钥
339
解密
341
337
342
删除
KS^{-1}_{SWA}
DOC
345
DOC
343
--SD1

图 22

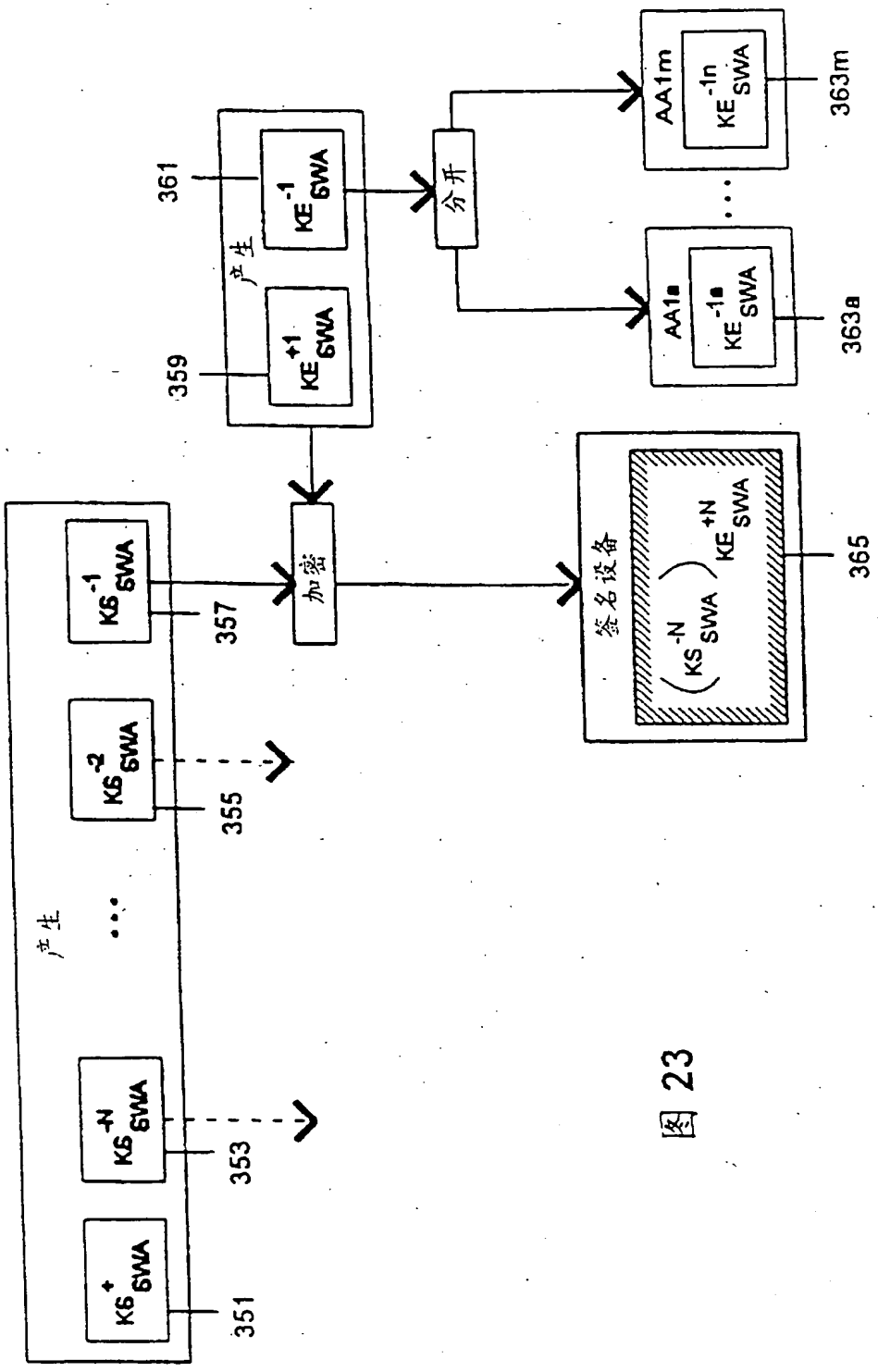

产生
KS^{+}_{SWA}
KS^{-N}_{SWA}
KS^{-2}_{SWA}
KS^{-1}_{SWA}
351
353
355
357
加密
产生
KE^{+1}_{SWA}
KE^{-1}_{SWA}
359
361
分开
AA1a
KE^{-1a}_{SWA}
363a
AA1m
KE^{-1n}_{SWA}
363m
签名设备
(KS^{-N}_{SWA}) KE^{+N}_{SWA}
365

图 23

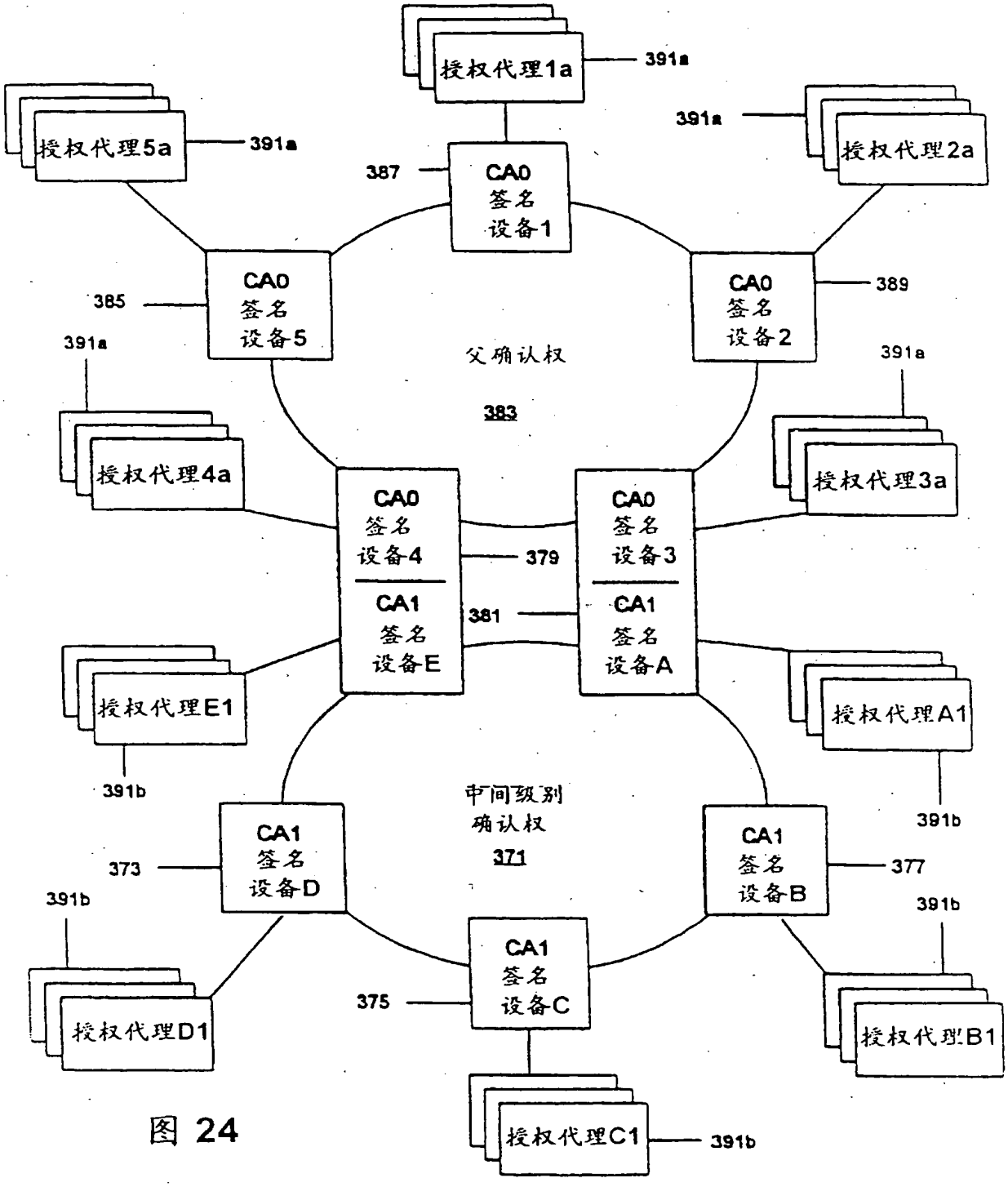

授权代理1a
391a
授权代理5a
391a
387
CA0
签名
设备1
391a
授权代理2a
CA0
签名
设备5
385
CA0
签名
设备2
389
391a
父确认权
383
391a
授权代理4a
授权代理3a
CA0
签名
设备4
379
CA0
签名
设备3
CA1
签名
设备E
381
CA1
签名
设备A
授权代理E1
授权代理A1
391b
391b
中间级别
确认权
371
CA1
签名
设备D
373
CA1
签名
设备B
377
391b
391b
CA1
签名
设备C
375
授权代理D1
授权代理B1
授权代理C1
391b

图 24

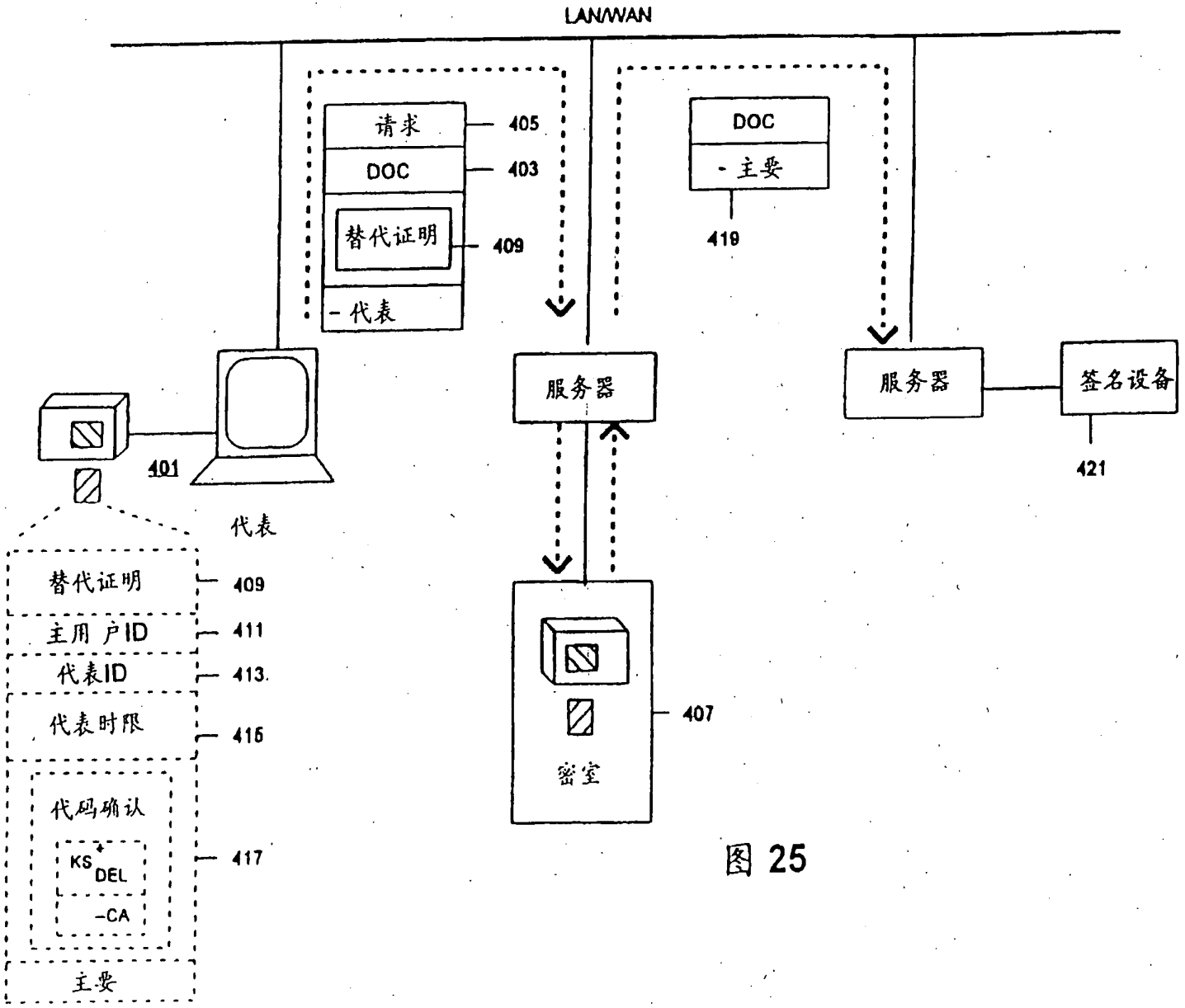

LAN/WAN
请求
405
DOC
403
替代证明
409
- 代表
DOC
- 主要
419
服务器
服务器
签名设备
421
401
代表
替代证明
409
主用户ID
411
代表ID
413
代表时限
415
代码确认
KS+DEL
-CA
417
主要
密室
407

图 25